Escola d'Enginyeria de Telecomunicació i Aeroespacial de Castelldefels
UNIVERSITAT POLITÈCNICA DE CATALUNYA

TRABAJO FINAL DE CARRERA

TÍTULO DEL TFC: Sistema de registro de defectos de calidad en asientos de automóvil

TITULACIÓN: Ingeniería Técnica de Telecomunicaciones, especialidad en Telemática

AUTOR: Rafael Rodríguez Cote

DIRECTOR: David Mota Jiménez

SUPERVISOR: Rafael Vidal Ferré

FECHA: 01 de Junio de 2015

Título: Sistema de registro de defectos de calidad en asientos de automóvil

Autor: Rafael Rodríguez Cote

Director: David Mota Jiménez

Supervisor: Rafael Vidal Ferré

Fecha: 01 de Junio de 2015

Resumen

En un mundo tan exigente como es el de la automoción, donde las grandes compañías luchan cada día por abarcar más cuota de mercado que su rival, hay un concepto que se potencia para obtener la máxima rentabilidad posible. Buscar la máxima calidad al menor coste posible.

El objetivo de este Trabajo Final de Carrera, realizado en la empresa Johnson Controls Eurosit Barcelona, es reforzar el sistema de detección y corrección de defectos de calidad en el producto, como parte de las políticas corporativas: la búsqueda de la excelencia y el incremento continuo de los estándares de calidad. Con ello se busca conseguir un sistema más fiable, robusto y preciso. Se pretende, además, mejorar el sistema de trazabilidad informatizándolo para disponer de un manejo de los datos en tiempo real mucho más flexible y adaptado a las necesidades de análisis de cada situación. Todo este conjunto de mejoras y modificaciones han de lograr, como objetivo final, reducir el número de asientos entregados al objetivo de la empresa de 0 defectos por asiento.

En la actualidad este proceso se realiza mediante la colocación de unos topos rojos sobre el asiento indicando el defecto. El conteo de los defectos es apuntado en una hoja por el operario que más tarde el auditor de calidad ha de pasar a Excel para su posterior explotación. Este sistema supone un gran número de operaciones manuales y el consecuente error humano asociado.

Para ello se propone la implementación de pantallas táctiles en el puesto de revisión con un listado de defectos de calidad según modelo/posición del asiento. La información del asiento se obtendrá por lectura de etiqueta de código de barras o conectando al PLC Siemens que gestiona las operaciones de montaje de la línea. Una vez registrados los defectos y almacenados en base de datos se mostrarán en un puesto de trabajo posterior de reparación y en otro adicional de verificación final. Además se mostrarán gráficas de defectos/hora/turno productivo en unas pantallas informativas instaladas a lo largo de la línea. Vía Intranet se podrán extraer informes de defectos y demás estadísticas.

Title: Quality defects registration system in car seats

Author: Rafael Rodríguez Cote

Director: David Mota Jiménez

Supervisor: Rafael Vidal Ferré

Date: June 1st 2015

Overview

In an industry such demanding as the automotive one, where all big companies compete on a daily basis to get a greater market share than their competitors, there is a principle underlying to achieving the maximum profitability possible: looking for the best quality at the lowest possible cost.

The aim of this project, performed at Johnson Controls Eurosit Barcelona Headquarters, is reinforcing the quality system for the detection and correction of defects in the finished product in order to fulfil some of the requirements stated on the corporate policies: the pursuit of excellence and the non-stop increasing of the quality standards.

This Project seeks to achieve a more reliable, robust and accurate quality system. It also intends to improve the computerized traceability system to provide data management in real time, adapted to the needs of each situation analysis.
All of the improvements and modifications are intended to achieve, as an ultimate goal, the reduction of the defects per seat to zero.

Nowadays, the quality procedure involves a large number of manual operations:
First, a few red moles are placed on the parts of the seat which are defective. After that, the operator records the defects data on a paper sheet. At last, the quality auditor introduces this information on the system.
This means the human element is a major factor when evaluating

In order to minimise the "human-error" factor considerably, this system proposes the implementation of touchscreens in the workstations where the review is performed. The values will be recorded using a predefined list of defects, according to the model of seat being inspected.

The seat information can be obtained from several ways:
It can be obtained directly by reading the barcode label on the seat or by connecting to the Siemens PLC that manages the Assembly line operations.

Once the defects are recorded and stored in the database, they will be displayed both in the subsequent repairing workstations and in the final verifications workstation. Additionally, the graphics of the production defects will also be shown all along the assembly line (they include the time and work shift where the defects have been produced).
The reports of the defects and other statistics can be removed via Intranet.

Agradecimientos

Aprovecho estas líneas para agradecer a todas las personas que me han ayudado a realizar este proyecto, sin el apoyo de todas ellos nada de esto habría sido posible.

En primer lugar quiero agradecer al supervisor de este proyecto, Rafael Vidal, por darme todas las facilidades posibles en la realización de este proyecto dentro de mi empresa.

En segundo lugar, a mi Director de proyecto y Manager, David Mota, por permitirme plasmar mi trabajo en este documento, guiarme en los momentos en los que no sabía cómo seguir y proporcionarme todas las herramientas y recursos que necesité para llevar a cabo este proyecto.

A Jordi Vilalta, por sentarse a mi lado, con infinita paciencia, y analizar paso a paso las direcciones de memoria del PLC hasta encontrar los fallos que yo era incapaz de ver.

A Oscar Caballer y Miguel Angel Díaz, por aguantarme 8 largas horas. Si no fuera por vosotros se harían interminables.

A los operarios, auditores de calidad, jefes de equipo y de taller y demás compañeros de trabajo a los que incordié en su día a día con mis pruebas, consultas y dudas. Vanessa, José, Rubén y Raquel. Muchas gracias por vuestra ayuda

Y por último, quiero agradecer de forma muy especial a los verdaderos artífices de que todo esto se haya puesto en marcha y haya concluido con éxito.
A mi hermana Raquel y mi abuela Paula, por estar siempre ahí cuando las necesito y tener una buena palabra en los labios para animarme.
A Lorena, mi sonrisa radiante, por apoyarme en todo momento y obligarme a pasar las tardes redactando estas líneas a pesar de los maravillosos planes alternativos que siempre se le ocurren. Por su paciencia y su forma de sacarme una sonrisa en cualquier situación.
A mis padres, Andrés y Mari Luz, por ser tan pesados y no dejar de insistir, año tras año, mes tras mes, día tras día, en que no dejará colgado el proyecto y acabara la carrera de una santa vez. Gracias por vuestra absoluta paciencia y por ser tan insistentes.

De corazón, muchas gracias a tod@s

ÍNDICE

INTRODUCCIÓN 1

CAPÍTULO 1. MOTIVACIÓN DEL PROYECTO 2

1.1. Johnson Controls, Inc. 2
1.1.1. Johnson Controls Eurosit 3
1.1.2. Metodología Just In Time 4
1.1.3. Estándares de calidad 6

1.2. Incidencias en el producto acabado 8
1.2.1. Escenario inicial e incidencia reclamada por el cliente 8
1.2.2. Solución propuesta al cliente 11
1.2.3. Fases de desarrollo 12

CAPÍTULO 2. ENTORNO DE TRABAJO 14

2.1. Infraestructuras y requerimientos hardware 14
2.1.1. PLC Siemens 15
2.1.2. Servidor OPC 16
2.1.3. Servidores Bases de Datos y Web 18
2.1.4. Terminales y escáneres de mano 19

2.2. Herramientas Software 21
2.2.1. Software de gestión de bases de datos 21
2.2.2. Entorno de programación Visual Studio 23
2.2.3. Microsoft IIS Server 24

CAPÍTULO 3. DESARROLLO DE LA HERRAMIENTA 25

3.1. Base de datos 29
3.1.1. Tablas 29
3.1.2. Procedimientos almacenados 32

3.2. Etiquetas de lectura OPC 33

3.3. Desarrollo de aplicaciones de escritorio 35
3.3.1 Funcionalidades comunes de las aplicaciones 35
3.3.2 Quality Checker 38
3.3.3 Quality Repair 42
3.3.4 Quality Firewall 44

3.4. Desarrollo de páginas web 47

CAPÍTULO 4. CONCLUSIONES 49

BIBLIOGRAFIA 51

ANEXOS 52

Anexo A. PLC Siemens 52

Anexo B. Base de datos 56

B.1. Creación de la base de datos 56
B.2. Tablas 58
B.3. Procedimientos almacenados 59

Anexo C. Creación de etiquetas de lectura OPC 61

Anexo D. Fichero de configuración 65

Anexo E. ASP Web Forms vs ASP.NET MVC 66

Anexo F. Modificaciones de desarrollo 70

INTRODUCCIÓN

En un mundo tan exigente como es el de la automoción, donde las grandes compañías luchan cada día por abarcar más cuota de mercado que su rival, hay un concepto que se potencia para obtener la máxima rentabilidad posible. Buscar la máxima calidad al menor coste posible.

El objetivo de este Trabajo Final de Carrera, realizado en la empresa Johnson Controls Eurosit Barcelona, es reforzar el sistema de detección y corrección de defectos de calidad en el producto, como parte de las políticas corporativas: la búsqueda de la excelencia y el incremento continuo de los estándares de calidad. Con ello se busca conseguir un sistema más fiable, robusto y preciso. Se pretende, además, mejorar el sistema de trazabilidad informatizándolo para disponer de un manejo de los datos en tiempo real mucho más flexible y adaptado a las necesidades de análisis de cada situación. Todo este conjunto de mejoras y modificaciones han de lograr, como objetivo final, reducir el número de asientos entregados al objetivo de la empresa de 0 defectos por asiento.

La redacción de esta memoria se ha estructurado en 4 capítulos claramente diferenciados:

CAPÍTULO 1. Motivación del proyecto: En este capítulo se explicarán las actividades de la empresa en la cual se ha desarrollado este TFC. Se tratará sobre sus estándares de calidad y los del cliente. A su vez se expondrá la problemática que motiva el desarrollo del sistema de registro de defectos realizado en este trabajo.

CAPÍTULO 2. Entorno de trabajo: En este segundo capítulo se describirá el entorno de trabajo en el que se ha desarrollado este TFC. Se analizaran, tanto las infraestructuras disponibles en la planta (servidores, PLCs, terminales y demás componentes hardware), como las herramientas software de las que se dispondrán. Así mismo se justificará adecuadamente el porqué del uso del hardware y software utilizado y no otro.

CAPÍTULO 3. Desarrollo de la aplicación: Este tercer capítulo contiene todos los aspectos técnicos acontecidos en el desarrollo de las herramientas de este TFC. En él se explican, tanto las funcionalidades implementadas, como las diferentes etapas en las que se han ido diseñado y poniendo en producción para cubrir las expectativas de rápida respuesta del cliente.

Por último esta memoria contiene un capítulo final en el que se recogen las conclusiones sobre el trabajo realizado y aspectos relacionados con el impacto medioambiental

CAPÍTULO 1. MOTIVACIÓN DEL PROYECTO

Este proyecto se ha realizado en la empresa Johnson Controls Eurosit, S.L. proveedor oficial de asientos de la planta de automoción SEAT Martorell. Johnson Controls Eurosit pertenece al área de negocio Automotive Experience de Johnson Controls Inc., uno de los mayores proveedores independientes de conjuntos de asientos del mundo.

En este capítulo se explicará quién es Johnson Controls Inc. y su planta producción de Abrera, cuál es su principal actividad económica y su relación comercial con su cliente SEAT. Se hablará sobre la metodología de producción JIT y los estándares de calidad de ambas empresas. Esto servirá para crear un escenario inicial que ayude al lector a comprender los conceptos básicos del mundo de la automoción y la problemática planteada cuya solución motiva la realización de este proyecto.

1.1. Johnson Controls, Inc.

La compañía Johnson Controls fue ideada en el año 1883 por Warren S. Johnson al conseguir la patente para realizar el primer termostato ambiental eléctrico, iniciándose así en la industria de los sistemas de control de edificios. En 1885, W. Johnson y un grupo de industriales de Milwaukee fundaron Johnson Controls Electric Service Company, con el objetivo de fabricar, instalar y servir sistemas automáticos de regulación de temperatura para edificios. En 1974, la compañía se rebautizó como Johnson Controls.

En los sucesivos años, la compañía se fue expandiendo tanto en mercado de negocio, como en ampliación de productos y servicios, hasta convertirse en una de grandes compañías mundiales en la experiencia de automoción, construyendo soluciones para eficiencia y energía.

En la actualidad, Johnson Controls posee más de 1.300 instalaciones repartidas en todo el mundo y alcanza aproximadamente la cifra de 17.0000 empleados.

La empresa se divide en tres unidades de negocio claramente diferenciadas:

- **Building Efficiency** (Soluciones para la eficiencia estructural): Es un importante suministrador de servicio completo de equipamiento mecánico, así como de sistemas de calefacción, ventilación, aire acondicionado y refrigeración, así como sistemas de seguridad para edificios.
- **Power Solutions** (Soluciones para energía): Es el mayor fabricante mundial de baterías de plomo para automoción y creador de procesos químicos avanzados para baterías.
- **Automotive Experience** (Soluciones para automóviles): Es el proveedor líder a nivel mundial en sistemas de interiores para vehículos ligeros

incluyendo automóviles de turismo y furgonetas. Los sistemas suministrados incluyen asientos, techos, puertas, paneles de instrumentación, alimentación y electrónica.

1.1.1. Johnson Controls Eurosit

J. C. Eurosit nace en Junio de 1990 al obtener el pedido para fabricar los asientos del modelo Ibiza de su cliente SEAT y comenzar a suministrarlos en septiembre del mismo año. Cinco semanas más tarde suministraba 500 coches diarios. El 15 de noviembre de 1990 J.C. Eurosit consiguió su primer “0” rechazos del cliente, manteniendo este nivel de servicio a lo largo de toda su trayectoria. En aquel entonces la factoría se encontraba ubicada en la Zona Franca de Barcelona.

Actualmente J.C. Eurosit se encuentra emplazada en la localidad de Abrera y tiene, como principal actividad económica, la fabricación de todos los asientos de los modelos de coche fabricados en la planta de automoción de SEAT Martorell. Además de la producción diaria demandada por SEAT, J. C. Eurosit también fabrica, en menor número, asientos especiales para modelos en desarrollo, salones comerciales, pruebas de explosionado de airbag y otro tipo de peticiones extraordinarias que no pertenecen a la secuenciación diaria recibida por el cliente.

Su producción se rige según la metodología JIT (Just In Time) siguiendo los estándares de calidad establecidos entre Johnson Controls Inc. y el Grupo Volkswagen.

Fig. 1.1 Vista aérea de la planta J. C. Eurosit Abrera

La planta dispone de 18.000 metros cuadrados 12.737 m^2 edificados de los cuales 11.677 m^2 están destinados al área productiva, y 1.060 m^2 a oficinas y servicios. La figura 1.2 muestra un diseño de la distribución de la planta en la que se pueden observar las 4 líneas de producción correspondientes, de arriba

abajo, a los modelos Ibiza, Q3, Q3 Posteriores, Q3 y León. En la parte izquierda del plano se encuentra el almacén y la zona de picking. En el extremo derecho la zona de expedición donde se almacenan los asientos acabados en contenedores secuenciados y se envían a SEAT.

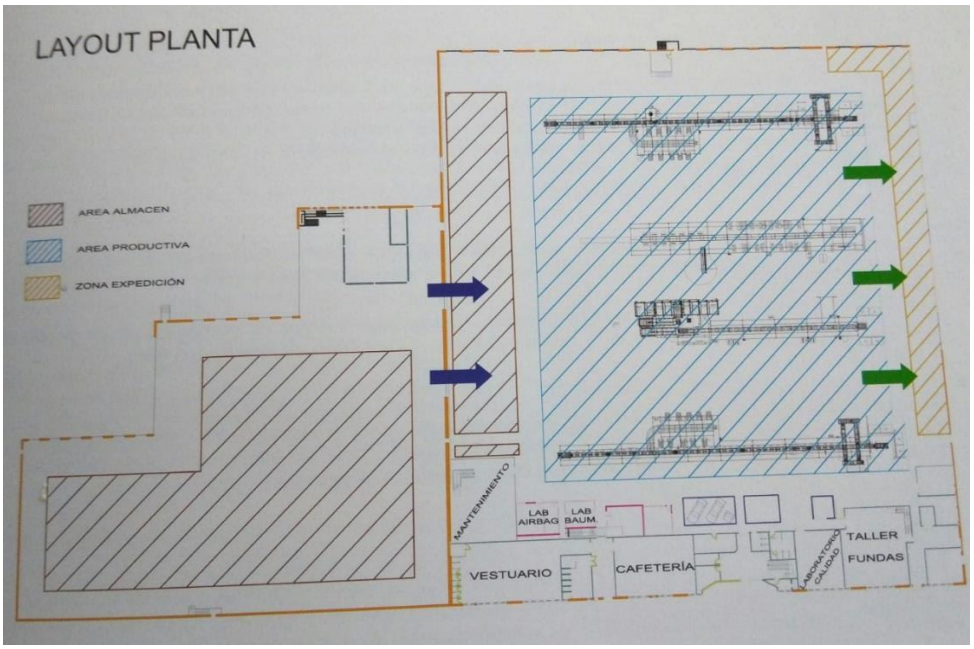


Fig. 1.2 Vista aérea de la planta J. C. Eurosit Abrera

1.1.2. Metodología Just In Time

La metodología JIT es un sistema de producción utilizado para mejorar la flexibilidad, productividad y calidad que se basa en producir justo lo que se necesita, en el momento que se necesita, con la mayor calidad posible y sin desperdiciar recursos del sistema.

Las bases de esta filosofía de trabajo son:

- Plazos de entrega cortos trabajando siempre bajo demanda.
- Eliminar el concepto de inventario, trabajar con stocks muy reducidos.
- Suprimir operaciones sin valor añadido tales como desplazamientos de personal o material innecesarios.
- Tolerancia cero a errores.
- Metodología de las 5S cuyo objetivo es disponer de puesto de trabajo organizados, ordenados, limpios y seguros. Las 5S corresponden a 5 principios básicos japoneses:
    1 – Seiri (Clasificar)
    2 – Seiton (Orden)
    3 – Seiso (Limpieza)
    4 – Seiketsu (Estandarizar)
    5 – Shitsuke (Disciplina)

La metodología JIT se atribuye a tres personas: el fundador de Toyota, Sakichi Toyoda, su hijo Kiichiro y el ingeniero Taiichi Ohno, quienes crearon este sistema entre 1946 y 1975 y supuso una revolución que impulso a Toyota a ser uno de los mayores fabricante mundiales de automóviles.

En la actualidad el sistema JIT es una formula muy extendida en varios ámbitos empresariales, con especial influencia en el sector de automoción. Johnson Controls y todas sus plantas de la división de automoción se rigen por estos principios.

La planta de SEAT Martorell se rige también por esta filosofía de trabajo, la cual extiende hacia sus proveedores. Periódicamente, mediante el sistema informático que SEAT comparte con sus proveedores, se envían unas previsiones de fabricación mensuales/semanales que permite planificar la gestión del mínimo stock de material para abastecer la producción diaria. A su vez, diariamente se envía la descripción de cada coche a fabricar. La planta de Eurosit recibe esa secuenciación y fabrica bajo demanda cada tipo de asiento solicitado. A un ritmo de producción normal sin ningún tipo de incidencia se dispone de un diferencial de 4-6 horas entre el momento en el cual el asiento sale de la línea de producción de Eurosit y se desplaza hasta el punto de montaje en la línea de montaje de SEAT. Esto supone una diferencia de 150-200 coches de margen con SEAT, lo cual implica que cualquier incidencia en la línea de producción puede afectar al cliente en un corto espacio de tiempo. A su vez, cualquier defecto o fallo en el producto acabado será detectado igual de rápido en el cliente. Este rápido ritmo de trabajo es lo que motiva que todos los sistemas de J. C. Eurosit funcionen con extrema precisión y tengan un sistema de rápida recuperación y detección de fallos. Los planes de contingencia y la rápida reacción ante cualquier tipo de incidencia son vitales para un proveedor JIT pues la detención de una línea de producción es un lujo que no se pueden permitir.

En la figura 1.3 se muestra de forma esquematizada el proceso que se acaba de comentar.

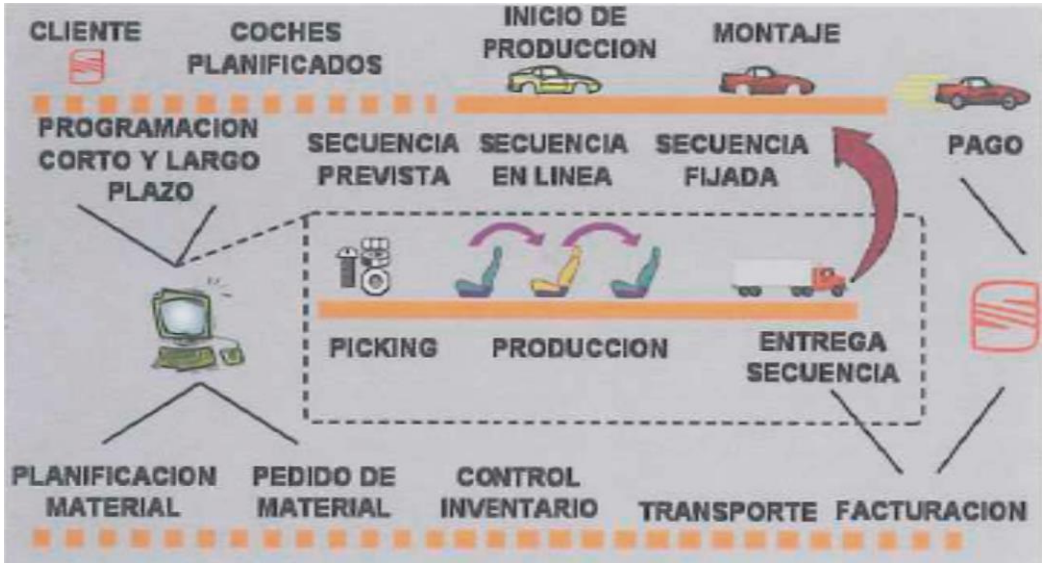


Fig. 1.3 Esquema de un proceso de fabricación JIT

1.1.3. Estándares de calidad

"Superar continuamente las crecientes expectativas de nuestros clientes."

En el manual del empleado que J. C. Eurosit distribuye a cada uno de sus trabajadores tras ser contratados, esta es la primera frase que se puede leer en el apartado "Política de calidad de Automotive Experience".

La política de calidad es algo muy importante para todo el conglomerado de Johnson Controls. El éxito económico de la empresa depende de la calidad sus productos pues éste es el principal valor añadido que Johnson Controls aporta para diferenciarse de sus competidores.

Las deficiencias cualitativas no sólo comportan desechos, reprocesados y suministros de reposición, sino que también pueden significar el pago de fuertes sumas económicas en concepto de indemnizaciones. Así mismo, las deficiencias cualitativas van ligadas a una pérdida de confianza por parte del cliente y a las consecuencias que eso conlleva. Es por eso que la Política de Calidad de Johnson Controls es muy estricta y está sujeta a constantes auditorias tanto por parte propia como del cliente.

Entre los diferentes procesos de calidad que se auditan dentro de J. C. Eurosit hay uno que prevalece por encima del resto. Se trata de la auditoria de Producto Acabado, en el cual se define el proceso requerido para valorar la calidad obtenida del producto acabado con el fin de poder observar la tendencia y la influencia de las acciones introducidas en el proceso / producto. La auditoría sirve para inspeccionar productos que están listos para el envío al cliente e identificar su cumplimiento con las especificaciones o características relevantes del mismo, hacer conclusiones de la calidad de las partes que se entregan, detectar el origen de las desviaciones e iniciar la implementación de medidas correctivas cuando sean necesarias. A cada característica que no cumpla con lo especificado se le atribuye una categorización según la gravedad del defecto detectado y se contabiliza con un valor establecido. Posteriormente se hace el recuento y se establece si la auditoria sobrepasa o no el Nivel de Calidad Admisible. Cuando el Nivel de Calidad Admisible sobrepasa el objetivo fijado por el Cliente, se deben decidir las acciones correctoras a aplicar sobre la producción, el producto terminado, y en tránsito. Si se observan fallos de la clase C, deben acordarse a corto plazo las medidas necesarias. En caso de defectos de categoría A, B y/o en una característica de seguridad, se han subsanar los defectos adoptando medidas inmediatas.

A modo de resumen para comprender la gravedad de cada tipo de defecto y las medidas adicionales que implican para la planta, con el consecuente incremento de costes, se anexan a continuación la tabla 1.1 donde se muestra la categorización de los diferentes tipos de defectos y las decisiones y medidas a tomar en cuenta.

Tabla 1.1. Clasificación de defectos, decisiones y medidas

Clase de Falla	Descripción de falla/ consecuencia	Medidas inmediatas	Medidas subsecuentes
A	El defecto lleva con toda seguridad a reclamaciones por parte del cliente. - Riesgo de seguridad, infracción de ley, paro funcional del coche - Producto no vendible o función no cumplida - Quejas extremas sobre la calidad de la superficie	- Bloquear y apartar las piezas existentes - Información a fábricas destinatarias y evaluación del riesgo - Medidas correctivas en el proceso de fabricación/ de verificación y en dado caso al 100%. - Medidas intensificadas de verificación en el proceso y en el producto terminado - En dado caso, verificación al 100% antes del suministro. Se requiere de una autorización para una desviación en el desarrollo –Demás medidas en mutuo acuerdo con la planta del cliente del Consorcio Volkswagen (ver Fórmula Q Concreta 4.4.1)	- Analizar más profundamente las actividades del proceso y de la verificación - Elaborar y llevar a la práctica medidas correctivas - Comprobación de la capacidad del proceso y del "cero defectos" - Comprobar la eficacia de las medidas iniciadas - En dado caso, iniciar un cambio de especificación
B	Se pueden esperar inconformidades del cliente o reclamaciones - Falla previsible - Utilidad disminuida		
C	Se pueden esperar reclamaciones de clientes exigentes. - Desviaciones que no influyen en el funcionamiento o en la operatividad - No disminuye la utilidad	- Información a las plantas receptoras para el acuerdo de medidas	

En la figura 1.4 que se muestra a continuación se pueden observar las categorías de las reclamaciones, escala de puntuación, repercusiones en la reacción del cliente y medidas a tomar.

Categoría de la reclamación	Reclamación A		Reclamación B		Reclamación C	
	A1	A	B1	B	C1	C
Escala de puntuación	140	80	60	40	20	10
Valoración de la reclamación	Riesgo para la seguridad del vehículo. Invencible. Cueca parada.	No aceptable, relacionado con la seguridad dará lugar a reclamaciones.	Son defectos más contrastables, son un claro alejamiento de los estándadres prescritos.	Defectos de calidad, incómodos, molestos y están fuera de los estándares prescritos.	Cabe esperar reclamaciones menos frecuentes pero más importantes.	Cabe esperar frecuentes reclamaciones de clientes exigentes.
Repercusiones en la reacción del clien	Vehículo no disponible.	El vehículo se debe llevar al taller antes de lo previsto.	El cliente hará subsanar el defecto en la próxima revisión del vehículo.		El cliente critica el nivel de calidad	
Antigua escala de reclamación	> 100	70	60 - 50	40 - 30	20	10
Detección	Por todos los clientes.					
	Por los clientes exigentes y el auditor teniendo en cuenta las medidas internas de la calidad.					
Eliminación del defecto	Se deberá eliminar el motivo de la reclamación: garantizar que ningún vehículo llegará al cliente con el defecto objeto de esta reclamación.					
Escala de puntuación	140	80	60	40	20	10
Medidas inmediatas	Bloquear y apartar las piezas existentes.				Información a las fábricas destinatarias, con el fin de acordar medidas.	
	Información a fábricas destinatarias y evaluación del riesgo.					
	Medidas correctoras en el proceso de fabricación / de verificación, en caso dado verificación total.					
	Medidas intensificadas de verificación en el proceso y en el producto acabado.					
	En caso dado, verificación total antes de la expedición.					
Medidas consecutivas	Analizar más profundamente las actividades del proceso y de la verificación.					
	Elaborar y llevar a la práctica medidas correctoras.					
	Dar prueba de la capacidad para el proceso y para los cero defectos.					
	Comprobar la eficacia de las medidas iniciadas.					
	En caso dado, iniciar un cambio de especificación					

Fig. 1.4 Categorías de las reclamaciones, escala de puntuación, repercusiones en la reacción del cliente y medidas a tomar

1.2. Incidencias en el producto acabado

En la exposición de los puntos anteriores se ha intentado introducir al lector en el complejo mundo de la automoción con el objetivo de, en este apartado, exponer las incidencias que pueden ocurrir durante el proceso de fabricación, las medidas a tomar y el desarrollo de éstas por fases para corresponder a las prioridades y exigencias del cliente.

1.2.1. Escenario inicial e incidencia reclamada por el cliente

Como ya se ha comentado anteriormente, J. C. Eurosit es el proveedor oficial de asientos de la planta de automoción de SEAT Martorell, encargándose pues de la fabricación de los modelos Ibiza, Q3 y León. Para ello dispone de 4 líneas de producción en las que se fabrican los asientos anteriores de Ibiza, Q3 y León y una adicional encargada de la fabricación de los asientos posteriores del modelo Q3. Tanto los asientos posteriores de Ibiza como los del León son fabricados por un proveedor subcontratado siendo de J. C. Eurosit tanto la licencia de fabricación como los procedimientos y estándares a aplicar.

Fig. 1.5 Línea de producción asientos anteriores León

Desde el momento en el que se inicia la fabricación de un asiento en el puesto de cabecera de línea hasta el momento en que se finaliza en el puesto de fabricación, éste circula por la línea de producción pasando por varios procesos de montaje y diferentes testeos de calidad y seguridad cada uno asociado a un puesto de montaje. En el momento en el que se inicia la fabricación de un asiento, toda la información del mismo en cuanto a requerimientos de montaje se asocia al palé en el que se montará. Este palé circulará por todos los puestos de montaje realizando las operaciones que requiera en cada uno de ellos y almacenando los datos de cada operación. Todas estas operaciones y datos obtenidos sobre cada asiento son gestionados por un PLC Siemens que se encarga de controlar toda la operativa de la línea de producción. Cada línea dispone de su propio PLC individual con sus correspondientes funciones y operaciones de montaje. Si bien es verdad que a grandes rasgos la operativa de montaje de todos los modelos es muy similar, cada uno de ellos se diferencia en pequeñas variaciones por su composición, siendo el Q3 el modelo más exigente y peculiar. La elaboración de este proyecto se ha realizado en torno a las líneas de producción de los modelos Ibiza y Q3, quedando la línea del León al margen al ser la de más nueva implantación. Esto ha supuesto que, por estándares de Johnson Controls, la herramienta que se ha diseñado en este TFC no se aplique al ya disponer ésta de una funcionalidad propia, la cual compararemos con la aquí expuesta en el apartado de conclusiones.

Por simplicidad, de ahora en adelante, en el desarrollo de esta memoria sólo se hará referencia a la línea del modelo Q3 a no ser que se indique lo contrario, puesto que es la línea que implica un mayor desarrollo y complejidad.

En la figura 1.6 se muestra el interfaz gráfico del PLC Siemens que gestiona la línea de producción de los asientos anteriores del Q3. En este caso la línea es circular. Se inicia en la Zona 1 y avanza hasta el puesto de expedición de la Zona 14 donde el asiento será almacenado en un contenedor y expedido hacia el cliente.

Previamente al punto anterior, una vez el asiento está completamente ensamblado, ha de pasar por el puesto de Inspección Visual en la Zona 11. En este puesto de trabajo, un auditor de calidad especializado en la revisión de todos los detalles y posibles defectos del asiento, verifica que el producto se adecúa a los estándares de calidad detallados en el apartado anterior y no presenta ningún tipo de defecto o desperfecto.

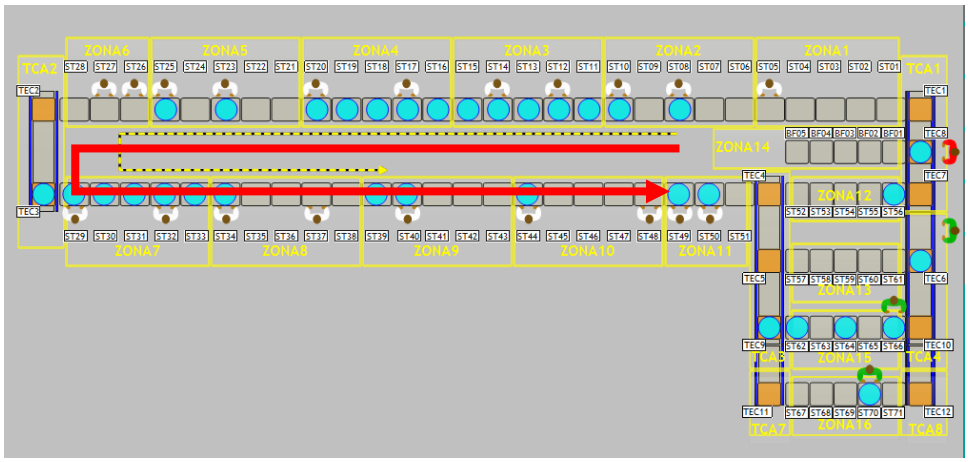


Fig. 1.6 Interfaz gráfico del PLC Siemens de la línea de anteriores del Q3

En el caso de estar en condiciones óptimas, el asiento es desviado hacia la Zona 14 donde será, como ya hemos comentado, almacenado y expedido. Sin embargo, si el inspector de calidad decide que el asiento tiene algún tipo de desperfecto lo indicará con un topo adhesivo de color (figura 1.7) y se desviará hacia las zonas 12, 13, 15 o 16 para su correspondiente retrabajo.

Fig. 1.7 Asiento anterior Q3 con dos defectos indicados con dos topos de color amarillo

En este punto es donde aparece la problemática y el motivo de la realización de este TFC. El procedimiento utilizado hasta el momento no es suficientemente robusto ni eficaz para las crecientes exigencias de calidad propias ni para las pactadas con el cliente. Indicar el defecto con un adhesivo de color rojo es ineficaz, puesto que el operario del bucle de retrabajo ha de revisar el asiento por completo en busca de defectos con el consecuente error humano que eso puede implicar. Es fácil pasar por alto defectos en zonas de difícil visualización como puede ser el cableado en la parte inferior del asiento para las regulaciones eléctricas. El tener que revisar de nuevo el asiento, a su vez, provoca un incremento considerable en el tiempo asignado al proceso. Y como consecuencia directa se produce un aumento en la ineficiencia de la línea. Por otro lado el registro de los defectos detectados es anotado a mano por el auditor de calidad en una hoja de papel, tal y como se muestra en la figura 1.8. Esto hace que el análisis de los defectos sea lento y muy tedioso. Las hojas se

recogen a final de turno para su posterior informatización por parte del departamento de calidad y análisis. Todo esto conlleva que defectos puntuales de una hora del día no sean detectados y analizados hasta semanas después, cuando el producto ya ha sido entregado.

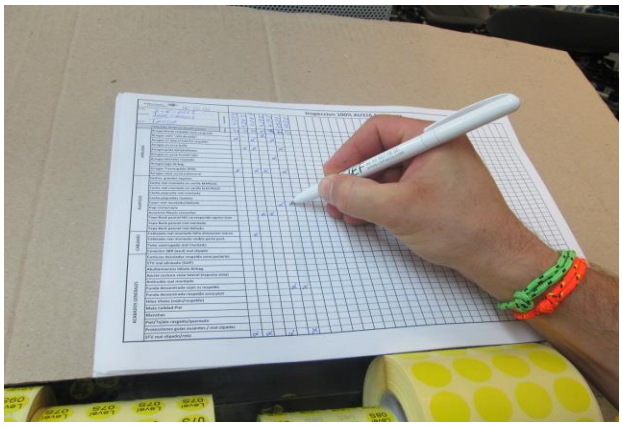


Fig. 1.8 Registro de defectos en asientos manual

Ante esta casuística, un número importante de defectos no son detectados y llegan al cliente, el cual los reporta y reclama actuaciones inmediatas para corregir este proceso.

1.2.2. Solución propuesta al cliente

Una vez detectado el problema reportado por el cliente, desde J. C. Eurosit se establece un plan de contingencia para solventar los errores de calidad detectados y establecer un nuevo proceso de revisión y reparación de defectos más fiable, robusto y eficaz.

En colaboración con los departamentos de Mantenimiento, Procesos, Producción y Calidad, se acuerda el desarrollo de una aplicación software en la que se registren los defectos detectados de cada asiento por los auditores de calidad y se almacenen en una base de datos evitando así la utilización de topos indicativos de difícil visualización. Esta aplicación se ejecutará en un terminal con una pantalla táctil instalado en el puesto de trabajo. La misma aplicación obtendrá la información de secuencia, línea de producción y demás detalles del asiento de forma automática, evitando así el error humano al escribir estos datos de forma manual. De esta forma se obtendrá un sistema robusto de información y se mejorará el tiempo de proceso del puesto de trabajo minimizando la tarea del auditor de anotación de defectos a un par de clics sobre una pantalla.

Una vez asociados estos defectos al asiento y almacenados, la aplicación marcará el asiento como no valido y será enviado hacia el bucle de rechazo donde se dispondrá de una segunda aplicación que mostrará la lista de defectos indicados por el auditor de calidad. Una vez reparados los defectos, el operador los marcará sobre la pantalla liberando el asiento y registrando que el defecto ha sido detectado y corregido. Toda esta información almacenada será registrada en una base de datos que quedará vinculada al resto de trazabilidad del asiento,

como son el código de airbag montado, la referencia de la funda o el ángulo y apriete de los tornillos.

Por último se decide incluir en la intranet de la compañía una aplicación web que permita la exportación de los datos almacenados según varios criterios de filtrado. Además se hace mucho hincapié sobre la necesidad de poder actuar rápidamente ante la detección de errores. Para ello se acuerda el montaje de pantallas a lo largo de las líneas de producción en las que se muestren estadísticas en tiempo real de los defectos detectados por el puesto de inspección visual. De esta forma, el jefe de línea, de un vistazo rápido a las pantallas de información, puede asociar el defecto a un puesto de trabajo en concreto y actuar al momento para evitar que se siga produciendo.

Todo esto conlleva un sistema robusto, eficaz y que permite actuar de forma rápida y eficiente sobre los defectos minimizándolos y mejorando los tiempos de proceso y la eficiencia de la línea, así como aumentando la productividad.

1.2.3. Fases de desarrollo

Tal y como se ha indicado en el apartado dedicado a los estándares de calidad y la metodología JIT, el trabajo en una planta de automoción ha de ser rápido, eficiente y con tolerancia 0 a los errores.

La detección por parte del cliente de defectos en el producto acabado no sólo exige la mejora del proceso de montaje, si no que establece unos plazos de actuación rápidos. Es por este motivo que el desarrollo de este TFC se ha realizado en varias etapas con el objetivo de cumplir con los plazos acordados y revisado mediante auditorias con el cliente.

Este TFC se ha realizado en 4 fases siendo la última un añadido final impuesto por la propia J. C. Eurosit con el objetivo de alcanzar su objetivo de 0 defectos en cliente.

Las fases de desarrollo son las expuestas en la siguiente planificación:

- **Fase 1 – Implementación mediante lectura de etiqueta. Modo escáner. Planificado para la semana 10.** Como ya se ha comentado el objetivo de este TFC es obtener la información del asiento a inspeccionar directamente de los datos almacenados en el PLC que gestiona la línea. La conexión de la aplicación a programar con este dispositivo es un tanto complicada. A fin de tener una solución parcial pero fiable dentro del plazo establecido, se decide omitir la conexión con el PLC y obtener los datos del asiento mediante el escaneo del código QR de la etiqueta con la que se identifica cada asiento, tal y como se puede observar en la figura 1.9. Se crea la primera versión de la base de datos con los campos básicos para almacenar los defectos asociados al asiento y un interfaz gráfico que muestre la información del asiento escaneado y el listado de defectos a revisar diseñado por el departamento de Calidad. Se crea la aplicación

del bucle de retrabajo siguiendo la misma filosofía de escaneado de etiqueta.

Fig. 1.9 Etiqueta de secuenciación de un asiento de la línea del Q3

- **Fase 2 – Implementación mediante lectura de PLC. Modo OPC. Planificado para la semana 16.** Se modifican las aplicaciones para conectar al PLC Siemens descartando así la lectura mediante pistola y mejorando el tiempo de proceso al evitar esta operación. Se rediseña el interfaz gráfico de la aplicación estructurando los botones de defectos según categorías sugeridas por el departamento de calidad y se depuran errores detectados por los operarios y auditores. Se modifica la base de datos para incluir nuevas funcionalidades como el registro del auditor que revisa los defectos, la ordenación y estructuración de los defectos según categorías para su fácil localización y se mejoran procedimientos estructurados para reducir el tiempo de proceso de algunas operaciones.

- **Fase 3 – Generación de informes y pantallas de información en tiempo real. Planificado para la semana 22.** Se crean diferentes páginas webs de estadísticas en tiempo real sobre los defectos, así como también un formulario para filtrar, según fechas, líneas y/o modelos y exportar a un informe Excel los defectos reportados según esos criterios. Se instalan pantallas de 27 pulgadas conectadas al servidor web mediante un terminal que muestran de forma periódica los datos en tiempo real de los defectos registrados.

- **Fase 4 – Desarrollo de una aplicación para el nuevo puesto de inspección final. Planificado para la semana 25**. Ante el buen funcionamiento del sistema y avanzándose a las nuevas exigencias del cliente respecto al modelo Q3, se decide añadir un nuevo puesto de trabajo situado tras el bucle de retrabajo. En este puesto se realiza una nueva verificación de los estándares de calidad y de posibles defectos no detectados ni por el auditor ni por el reparador. Con el objetivo de registrar que todos los asientos han pasado por una segunda revisión y anotar si se han encontrado y corregido defectos no detectados en los puestos anteriores, se diseña una nueva aplicación que disponga de estas funcionalidades. Se modifica la base de datos para incluir los datos de esta nueva aplicación y se incluyen sus estadísticas en los informes en tiempo real de las pantallas de información.

CAPÍTULO 2. ENTORNO DE TRABAJO

2.1. Infraestructuras y requerimientos hardware

Una vez analizada la problemática de la planta y realizada la propuesta de solución, se empezará a trabajar en la fase de diseño e implementación de las herramientas antes mencionadas. Para ello se analizará las infraestructuras y el hardware disponible en la planta con el que se debe de interactuar.

Tal y como se puede observar en la figura 2.1, el objetivo es interactuar con la línea de producción mediante un terminal con pantalla táctil ubicado en el puesto de montaje. Para ello se dispondrá de un servidor OPC que establecerá las comunicaciones entre el PLC que gestiona la línea y el terminal en el que interactúa el operario.

Por otro lado, el registro de todos los defectos detectados por el auditor de calidad ha de ser almacenado en una base de datos para, más adelante, consultarse en el puesto de reparación. El terminal, pues, también ha de conectarse con el Servidor de Base de Datos.

Finalmente, la figura muestra las pantallas de información conectadas al servidor web para así poder mostrar las diferentes páginas webs implementadas con la información en tiempo real para los jefes de línea.

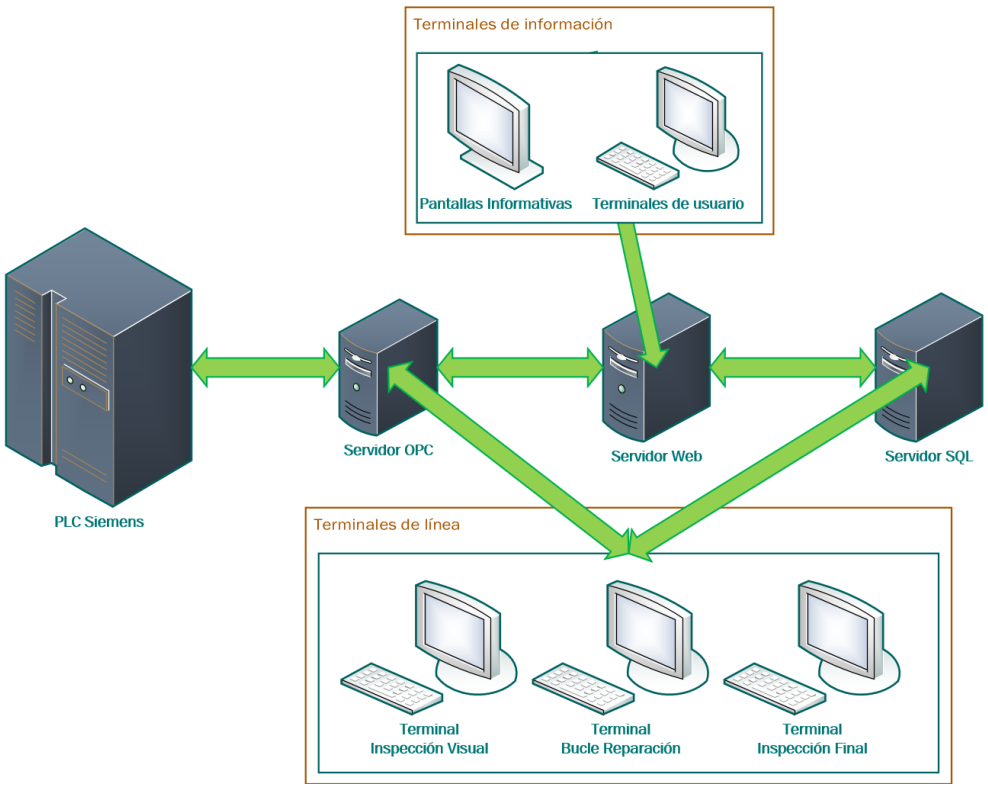


Fig. 2.1 Diagrama de componentes hardware para la solución implementada en el TFC

2.1.1. PLC Siemens

Como ya se ha comentado en varias ocasiones, la planta de J. C. Eurosit dispone de 4 líneas de montaje, cada una de ellas gestionadas por un PLC Siemens. Un PLC (Programamable Logic Controller) no es más que un equipo electrónico diseñado para controlar procesos secuenciales en tiempo real.

Un PLC es un equipo comúnmente utilizado en maquinarias y líneas de montaje industriales. En el anexo A, incluido en la parte final de esta memoria, se expondrán las principales características de un PLC así como se mostrarán los modelos con los que se trabajan en J. C. Eurosit con más detalle. En este apartado se describirá la parte del PLC más relacionada con el desarrollo de este TFC, los bloques de memoria o DBs.

La programación de un PLC Siemens S7 – 300 se compone de tres bloques básicos:

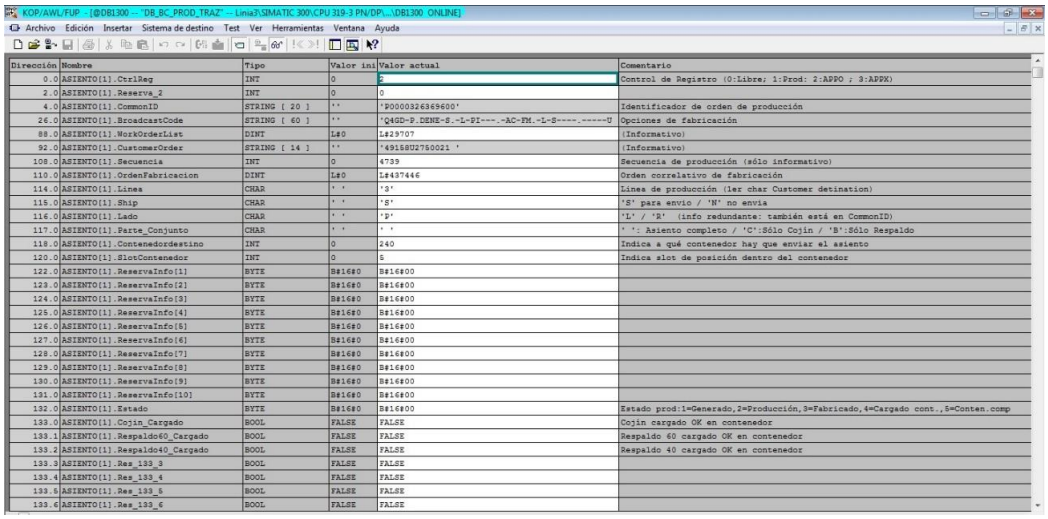
- Bloques de organización (OB)
- Bloques de función (FB)
- Bloques de memoria (DB)

Los bloques de memoria o DBs, son áreas de datos en las que se almacena la información de la línea de producción. En ellos se incluyen todos los datos de cada uno de los asientos, así como la relación de acciones efectuadas para cada asiento en proceso de montaje. El desarrollo de este TFC requiere que se acceda a las DBs adecuadas para obtener la información de los asientos en los puestos de inspección de calidad y retrabajo. De esta forma se obtendrá una información fiable del asiento a inspeccionar evitando el error humano en el momento de registrarlo en la base de datos.

Para el desarrollo de esta aplicación necesitaremos obtener la siguiente información de cada asiento:

- **Línea de producción.** Entero que identifica la línea del Ibiza (1), León (2) y Q3 (3)
- **Número de secuencia.** Entero de cuatro dígitos que identifica a un coche. Al llegar al 9999 se reinicia empezando por el 0001
- **Mano.** Carácter ‘L’ (Izquierda) o ‘R’ (Derecha) que indica si el asiento es el derecho o el izquierdo.
- **VIN Code.** Identificador alfanumérico que identifica un coche de forma única.
- **LBHD.** Identificador alfanumérico que identifica un asiento de forma única.
- **Broadcast Code.** Combinación de letras y números que definen toda la estructura y opciones de configuración de un asiento.
- **Fuera Secuencia.** Carácter que establece si el asiento pertenece a la secuenciación de SEAT (‘N’) o por el contrario es un montaje excepcional fuera de la secuenciación de SEAT (‘S’)

En la figura 2.2 que se muestra a continuación se visualiza, en tiempo real, una estructura DB y los valores almacenados en cada una de ellas.



Dirección	Nombre	Tipo	Valor ini	Valor actual	Comentario
0.0	ASIENTO[1].CtrlReg	INT	0	2	Control de Registro (0:Libre; 1:Prod: 2:APPO ; 3:APPX)
2.0	ASIENTO[1].Reserva_2	INT	0	0	
4.0	ASIENTO[1].CommonID	STRING [20]	''	'P000032639600'	Identificador de orden de producción
26.0	ASIENTO[1].BroadcastCode	STRING [60]	''	'Q4GD-P.DENE-S.-L-PI---.-AC-FM.-L-S----.-----U	Opciones de fabricación
88.0	ASIENTO[1].WorkOrderList	DINT	L#0	L#29707	(Informativo)
92.0	ASIENTO[1].CustomerOrder	STRING [14]	''	'4915SU2750021 '	(Informativo)
108.0	ASIENTO[1].Secuencia	INT	0	4739	Secuencia de producción (sólo informativo)
110.0	ASIENTO[1].OrdenFabricacion	DINT	L#0	L#437446	Orden correlativo de fabricación
114.0	ASIENTO[1].Linea	CHAR	' '	'3'	Línea de producción (1er char Customer detination)
115.0	ASIENTO[1].Ship	CHAR	' '	'S'	'S' para envio / 'N' no envia
116.0	ASIENTO[1].Lado	CHAR	' '	'D'	'L' / 'R' (info redundante: también está en CommonID)
117.0	ASIENTO[1].Parte_Conjunto	CHAR	' '	' '	' ': Asiento completo / 'C':Sólo Cojín / 'B':Sólo Respaldo
118.0	ASIENTO[1].Contenedordestino	INT	0	240	Indica a qué contenedor hay que enviar el asiento
120.0	ASIENTO[1].SlotContenedor	INT	0	5	Indica slot de posición dentro del contenedor
122.0	ASIENTO[1].ReservaInfo[1]	BYTE	B#16#0	B#16#00	
123.0	ASIENTO[1].ReservaInfo[2]	BYTE	B#16#0	B#16#00	
124.0	ASIENTO[1].ReservaInfo[3]	BYTE	B#16#0	B#16#00	
125.0	ASIENTO[1].ReservaInfo[4]	BYTE	B#16#0	B#16#00	
126.0	ASIENTO[1].ReservaInfo[5]	BYTE	B#16#0	B#16#00	
127.0	ASIENTO[1].ReservaInfo[6]	BYTE	B#16#0	B#16#00	
128.0	ASIENTO[1].ReservaInfo[7]	BYTE	B#16#0	B#16#00	
129.0	ASIENTO[1].ReservaInfo[8]	BYTE	B#16#0	B#16#00	
130.0	ASIENTO[1].ReservaInfo[9]	BYTE	B#16#0	B#16#00	
131.0	ASIENTO[1].ReservaInfo[10]	BYTE	B#16#0	B#16#00	
132.0	ASIENTO[1].Estado	BYTE	B#16#0	B#16#00	Estado prod:1=Generado,2=Producción,3=Fabricado,4=Cargado cont.,5=Conten.comp
133.0	ASIENTO[1].Cojin_Cargado	BOOL	FALSE	FALSE	Cojín cargado OK en contenedor
133.1	ASIENTO[1].Respaldo60_Cargado	BOOL	FALSE	FALSE	Respaldo 60 cargado OK en contenedor
133.2	ASIENTO[1].Respaldo40_Cargado	BOOL	FALSE	FALSE	Respaldo 40 cargado OK en contenedor
133.3	ASIENTO[1].Res_133_3	BOOL	FALSE	FALSE	
133.4	ASIENTO[1].Res_133_4	BOOL	FALSE	FALSE	
133.5	ASIENTO[1].Res_133_5	BOOL	FALSE	FALSE	
133.6	ASIENTO[1].Res_133_6	BOOL	FALSE	FALSE	

Fig. 2.2 Visualización de los valores almacenados en bloques de memoria

2.1.2. Servidor OPC

Como se ha visto en el apartado anterior, el PLC dispone de toda la información que requieren las aplicaciones. El objetivo es acceder a las posiciones de memoria adecuadas para obtener los datos del asiento a revisar tanto en el puesto de inspección de calidad como en el bucle de retrabajo.

Acceder directamente a las DB del PLC desde una aplicación de escritorio no es algo trivial. Si bien es verdad es que existen soluciones que permiten, mediante el uso de librerías externas, manejar datos del PLC directamente desde el código de la aplicación, no es un sistema que se utilice en la compañía.

Las políticas internas de Johnson Controls se encaminan a minimizar el número de conexiones directas hacia el PLC para evitar sobrecargar el sistema. En base a ello, se utiliza desde hace años un servidor OPC para establecer las comunicaciones entre PLC y aplicación cliente. Este servidor centraliza todas las conexiones al PLC permitiendo así un control global desde un único punto de acceso. A su vez permite una fácil adaptación de cualquier variación en los procesos del PLC que de otra forma implicaría modificaciones individuales más costosas en tiempo y dinero.

OPC (OLE for Process Control) es un estándar de comunicación en el campo del control y supervisión de procesos industriales, basado en una tecnología de Microsoft, que ofrece un interfaz común para comunicación que permite que componentes software individuales interactúen entre sí y compartan datos. La comunicación OPC se realiza a través de una arquitectura Cliente-Servidor. El servidor OPC es la fuente de datos, vinculado a los dispositivos que queramos acceder como PLCs, atornilladoras y demás automatismos de las líneas de producción. A través de él, cualquier aplicación basada en OPC puede acceder a dicho servidor para leer/escribir cualquier variable que ofrezca el servidor. Es una solución abierta y flexible que prácticamente todos los mayores fabricantes

de sistemas de control, instrumentación y de procesos han incluido en sus productos.

En el caso de J. C. Eurosit se dispone de un servidor dedicado a alojar el servicio OPC conectado a todas los PLCs y atornilladoras de la planta. Actualmente se dispone del software ToolBox OPC Power Server en su versión 5.2.133.0 del año 2010.

La instalación del software crea un servicio de escucha en el servidor mediante el cual las diferentes aplicaciones que programemos intercambiaran datos con este. Por otro lado, dentro del software de gestión se deben crear las conexiones a las DBs de PLC que interesen y nombrarlas con un identificador único o etiqueta. De esta forma la aplicación cliente solicitará los datos de una etiqueta en concreto y el servidor responderá devolviendo el valor de dicha etiqueta en ese momento.

En la figura 2.3 se muestra un ejemplo del programa de gestión que se utilizará. En la parte izquierda de la imagen se muestra la estructura jerárquica de las diferentes etiquetas configuradas en el sistema. La parte derecha de la imagen muestra una serie de etiquetas configuradas con sus principales características.

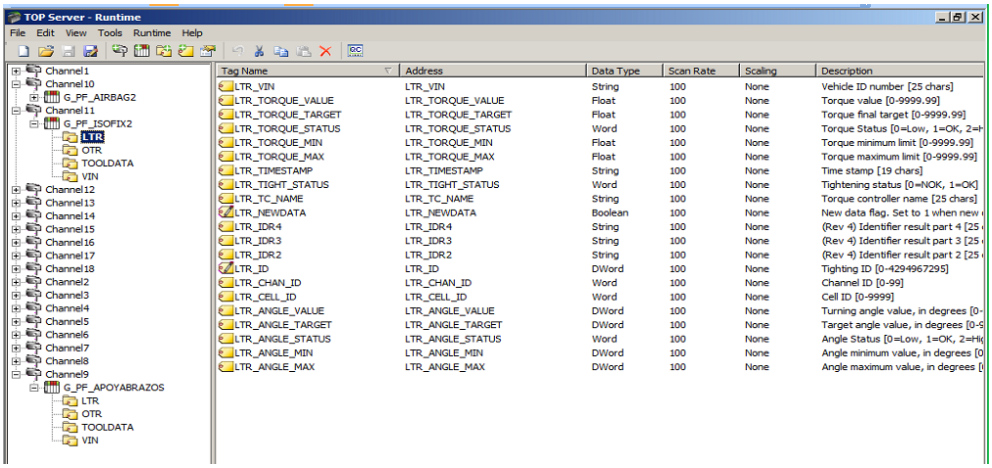


Fig. 2.3 Visualización de los valores almacenados en bloques de memoria

Por el momento se obviará como se configuran las diferentes etiquetas que se han empleado para obtener la información que necesitamos del PLC. Esto será explicado con más detalle en el siguiente capítulo.

Si bien en este TFC se va a usar este servidor dedica OPC para establecer la comunicación entre PLC y aplicación cliente, hay otro tipo de soluciones, como ya se ha comentado antes, que permiten prescindir de este servidor intermedio con labores de intérprete. En J. C. Eurosit se está empezando a utilizar en los nuevos desarrollos realizados una serie de librerías, llamadas libnodave, que permiten comunicar aplicaciones programadas en VB.Net, Java, C y C++ con los PLC Siemens S7 - 300 y S7 – 400. A pesar de que el uso de estas librerías supone un ahorro en cuanto a infraestructuras y mantenimiento, los desarrollos que se han realizado por el momento son bastante básicos y aún se encuentran

en fase de pruebas. Por las exigencias de tiempo y fiabilidad que se definieron en la elaboración de este TFC se ha descartado el uso de la librería libnodave.

2.1.3. Servidores Bases de Datos y Web

Uno de objetivos en la elaboración de este TFC es el de almacenar de forma fiable toda la información de los defectos detectados y corregidos mediante las aplicaciones a desarrollar así como su explotación y manejo de forma rápida y eficiente.

Para almacenar la información de todos sus procesos de trazabilidad de asientos, producciones, incidencias en línea y demás datos importantes J. C. Eurosit dispone de dos servidores de bases de datos separados según la información que manejen cada uno. Así pues se dispone de un servidor crítico, llamado así pues su almacena información relacionada con las líneas de producción y sus procesos. Es por ello que su funcionamiento está vinculado a las líneas de producción y cualquier fallo de servicio podría ocasionar un paro en la producción o sus procesos. El segundo servidor almacena otro tipo de información no vinculada directamente con la producción por lo cual su funcionamiento no afecta al trabajo en las líneas de montaje.

Por la naturaleza de la aplicación a desarrollar los datos serán almacenados en una nueva base de datos creada en el servidor crítico. En el próximo capítulo se analizará la estructura de dicha base de datos así como sus tablas, procedimientos almacenados y demás decisiones de diseño realizadas.

Se dispondrá también de un servidor web en el que alojar los informes a tiempo real de los defectos detectados y corregidos.

Johnson Controls tiene un acuerdo global de licencia con Microsoft para todos sus dispositivos y servidores, lo cual implica que, exceptuando algunos servidores específicos, los sistemas operativos de sus servidores son Windows Server, se trabaja con Microsoft SQL Server y las páginas webs se alojan en Internet Information Services (IIS). Éste es el factor diferencial que motiva el desarrollo de este TFC utilizando lenguajes de programación creados por Microsoft, tales como Visual Basic.Net y ASP.Net MVC.

Desde hace unos años, la política de infraestructuras de J. C. Eurosit se ha decantado por virtualizar todos sus servidores. Actualmente se dispone de tres grandes servidores físicos idénticos sobre los cuales se virtualizan todos los servidores individuales correspondientes a las diferentes necesidades de negocio de la planta a excepción de dos servidores físicos dedicados a almacenamiento de ficheros y copias de seguridad y un tercer servidor UNIX encargado de la secuenciación de los asientos recibidos por el cliente. En la figura 2.4 se pueden observar los detalles de estos tres servidores y las máquinas virtuales que cada uno aloja. Algunos de los datos han sido parcialmente borrados para preservar la seguridad de los sistemas de J. C. Eurosit.

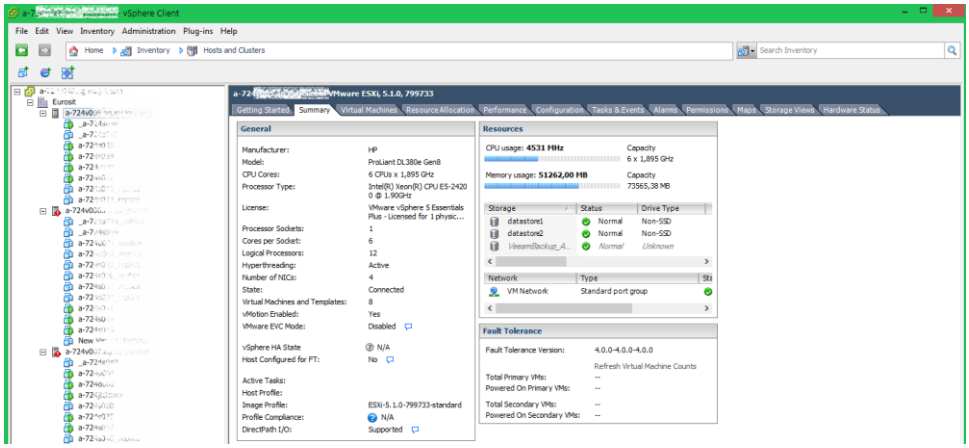


Fig. 2.4 VMware vSphere Client visualizando la estructura de servidores virtuales de J. C. Eurosit

Fig. 2.5 Rack de servidores del CPD Principal de Eurosit

2.1.4. Terminales y escáneres de mano

Finalmente, para concluir con este apartado sobre las infraestructuras y dispositivos hardware empleados en la elaboración de este TFC se hablará sobre los terminales instalados en los puestos de trabajo.

Para poder ejecutar las aplicaciones desarrolladas no se requerirá un ordenador de gran potencia y rendimiento. Con un terminal sencillo, con conexión Ethernet, pantalla táctil y conexión USB a la cual conectar el escáner de mano de códigos de barra tendremos suficiente.

La solución que Johnson Controls propone a sus plantas para todos estos terminales sencillos en los puestos de trabajo son los llamados Thin Clients de HP. Tal y como se observa en la figura 2.6 son pequeñas estaciones de trabajo preparadas para entornos industriales, de bajo consumo y gran rendimiento.

Actualmente en J. C. Eurosit se trabaja con los modelos HP t510 y t610, aunque comenzarán a sustituirse por los nuevos modelos t520 y t620 en breve.

El Thin Client t610 utiliza chipset AMD con unidad de procesamiento acelerado APU G-Series, compuesto de micro de doble núcleo a 1,65 GHz y gráfica Radeon. En cuanto al HP t510 utiliza un chip VIA Eden U4200, con micro de doble núcleo a 1 GHz y gráficos VIA Chromotion HD 2.0 con soporte para aceleración por hardware. Ambos incorporan 2 Gbytes de memoria RAM.

Con respecto a las pantallas táctiles se han utilizado pantallas ELO TouchScreen de 15 pulgadas panorámicas modelo ET1509L-8UWA.

En la figura 2.7 vemos un conjunto del terminal de trabajo montado en el puesto de inspección de calidad con la aplicación ejecutándose.

Fig. 2.6 Thin Clients HP

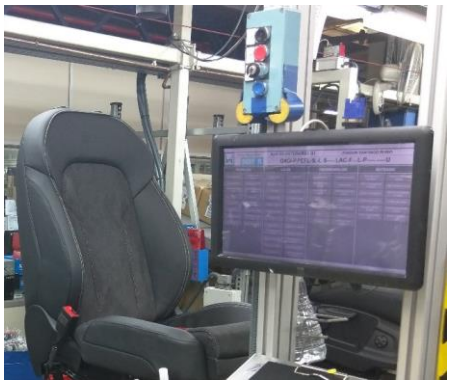


Fig. 2.7 Puesto de trabajo

Como se comentó en el capítulo 1 de este TFC, en la fase inicial del desarrollo del proyecto, debido al corto plazo de tiempo para adoptar una solución satisfactoria para el cliente, se decidió utilizar un escáner de mano lector de códigos QR para obtener la información de la etiqueta de secuenciación (figura 2.8) y no del PLC. El modelo utilizado por J. C. Eurosit corresponde a una Datalogic Powerscan PM9500 idéntica a la que se muestra en la figura 2.9

Fig. 2.8 Etiqueta de secuenciación

Fig. 2.9 Escáner PM9500

Por último, en la figura 2.10, se muestran las pantallas TFT de AOC de 27 pulgadas distribuidas a lo largo de la línea. En ellas se muestran las estadísticas en tiempo real de los defectos detectados en el puesto de inspección de calidad, así como los datos de producción del turno actual y el stock de seguridad disponible con el cliente.

Fig. 2.10 Pantallas de información a tiempo real en las líneas de producción

2.2. Herramientas Software

Una vez definidas las infraestructuras y dispositivos hardware con los que se trabajará en este TFC se pasará a explicar la parte software comentando las principales herramientas que utilizaremos en el desarrollo de nuestro proyecto.

Este apartado se dividirá en tres subapartados correspondientes a las aplicaciones empleadas para el diseño y la gestión de las bases de datos, el IDE de desarrollo para las aplicaciones cliente así como para las páginas web, enlazando estas últimas con el entorno web en el que se aloja la intranet de J. C. Eurosit.

2.2.1. Software de gestión de bases de datos

Como ya se ha comentado en el apartado referente al servidor de base de datos, J. C. Eurosit trabaja con Microsoft SQL Server. Concretamente con su versión 2008. Para la creación de bases de datos, tablas, procedimientos almacenados y demás funcionalidades y configuraciones de la base de datos, J. C. Eurosit dispone de dos herramientas, SQL Server Managment Studio y TOAD for SQL Server.

La primera de ellas corresponde a la aplicación nativa de Microsoft para acceder, configurar, administración y desarrollar bases de datos. Dispone a su vez herramientas para el análisis de consultas, rendimiento de procedimientos y generación de script.

Si bien es la herramienta oficial de Microsoft para trabajar con base de datos, a la hora de desarrollar y realizar tareas rutinarias, gestiones de datos entre bases de datos y otros tipos de análisis de desarrollo, la herramienta es poco flexible y funcional.

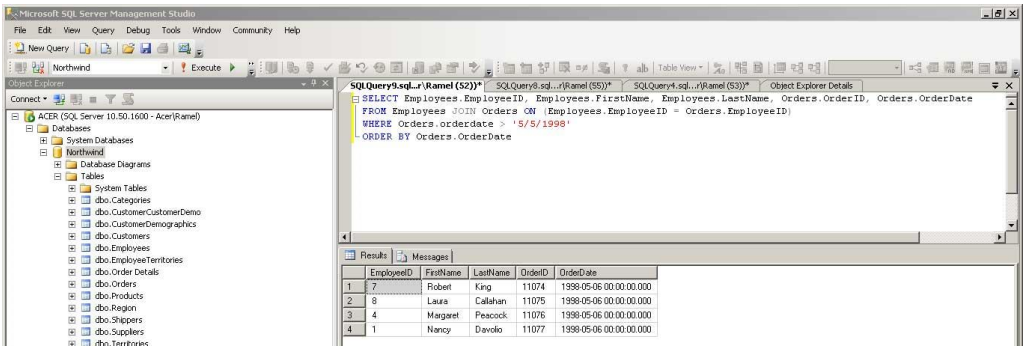


Fig. 2.11 Ejemplo de entorno SQL Managment Studio

Es por eso que en J. C. Eurosit se trabaja con otro software de desarrollo y administración de bases de datos. TOAD nació como un administrador de bases de datos Oracle resultando una herramienta muy útil para los administradores y desarrolladores de bases de datos. Su uso se fue extendiendo y actualmente está disponible para bases de datos Oracle, Microsoft SQL Server, IBM DB2 y MySQL.

Esta herramienta no solo permite las funcionalidades básicas de creación de bases de datos, tablas y procedimientos almacenados, sino que otorga una gran potencia de análisis de bases de datos, comparativas de datos y definiciones entre tablas o instancias diferentes y la generación automática de un sinfín de scripts de modificaciones y/o actualizaciones de datos.

A la hora de trabajar con SQL en este TFC se usará preferentemente TOAD for SQL para el desarrollo, administración y gestión de las bases de datos, relegando así SQL Managment Studio para funciones de administración y configuración de los servicios SQL del servidor.

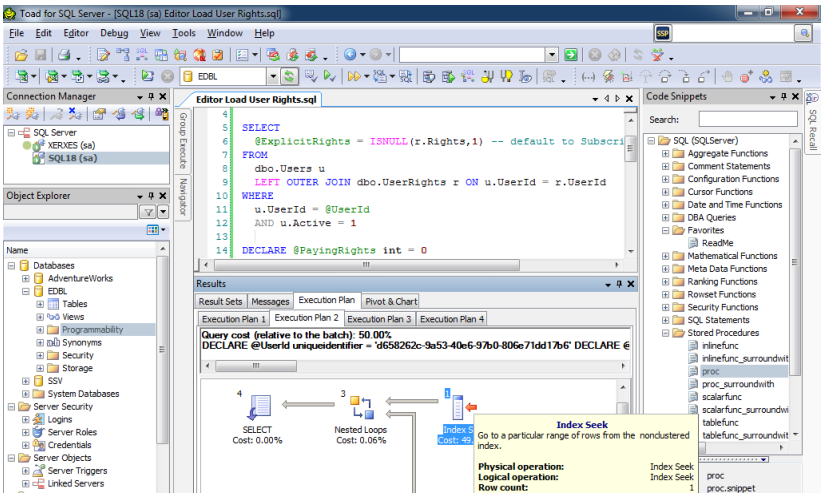


Fig. 2.12 Ejemplo de entorno TOAD for SQL Server

2.2.2. Entorno de programación Visual Studio

Debido al acuerdo global existente entre Johnson Control y Microsoft ya indicado en capítulos anteriores, se ha descartado la utilización de lenguajes como Java, de código abierto, que si bien ofrecen un menor coste a la hora de no requerir el pago por licencias de uso, no permiten una característica importante que sí ofrece este entorno: una compatibilidad mayor con todo el universo de aplicaciones de Johnson Controls.

Microsoft Visual Studio es un IDE (entorno de programación) para sistemas operativos Windows. Permite el desarrollo de aplicaciones de escritorio y aplicaciones, sitios web y servicios web en cualquier entorno de trabajo que soporte la plataforma .NET. Soporta multiples lenguajes de programación tales como ASP, ASP.NET MVC, C++, C#, PHP, Ruby y VB.NET entre otros.

Este entorno permite desarrollar las aplicaciones de escritorio requeridas para la realización de este TFC así como la creación de las páginas web de información en tiempo real y de extracción de datos. A su vez, la conexión de las aplicaciones con las base de datos SQL server se establece de forma muy sencilla, permitiendo así un entorno de programación global y unificado.

Por último, Visual Studio soporta la utilización de librerías externas que ayudan a integrar funcionalidades de otros sistemas en el código. En el desarrollo de este TFC se usa la librería XmlDANet.dll propia del servidor OPC. Mediante esta librería se puede acceder a las funciones ya programadas que ayudan a conectarse al servidor OPC y obtener la información de las etiquetas que sean necesarias.

En la figura 2.13, se observa una imagen del entorno de programación Visual Studio en su versión 2013 con el código de la aplicación Quality Checker desarrollada. En la barra izquierda de herramientas se muestran los diferentes componentes que se pueden añadir a un formulario de VB.Net. En la parte central superior se observa el diseño gráfico del formulario con su correspondiente código en la parte inferior. Por último en la parte derecha se encuentra la estructura del proyecto y bajo éste las propiedades de elementos.

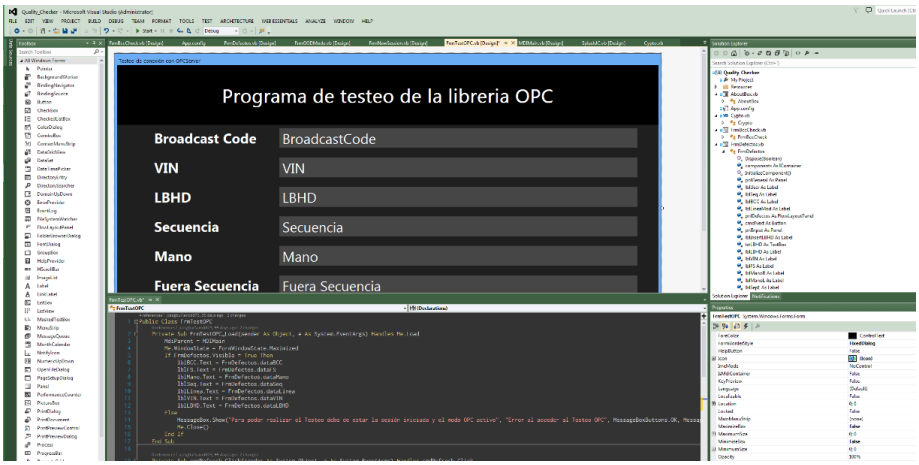


Fig. 2.13 Ejemplo de entorno de programación Visual Studio 2013

2.2.3. Microsoft IIS Server

La última herramienta fundamental para la elaboración de este proyecto es el servidor web para servidores Microsoft. Internet Information Service o IIS es el conjunto de herramientas y servicio que permiten crear un servidor web compartiendo información con usuarios en Internet, ya sea en una intranet o en una extranet.

IIS ofrece servicios de HTTP, FTP, SNMP y NNTP y se basa en varios módulos que le dan capacidad para procesar diferentes tipos de página, ya sean las nativas ASP y ASP.NET o pertenecientes a otros estándares como puede ser PHP

En J. C. Eurosit se dispone, como ya se ha comentado anteriormente, de un servidor dedica en el cual se aloja la versión 7 de IIS sobre el cual se alojan tres intranets propias.

Actualmente el equipo de IT se encuentra en plena proceso de migración de la antigua Intranet, la cual está programada en ASP Web Forms, a una nueva y mejorada versión que utiliza la tecnología ASP.NET MVC.
Por el momento, ambas intranets conviven mientras se van migrando las diferentes aplicaciones que componen el global de servicio de intranet de J. C. Eurosit.

En cuanto a la tercera intranet mencionada, no es más que un entorno de preproducción, creado especialmente para este proceso de migración. Una vez la aplicación a migrar ha sido diseñada y probada en el entorno de desarrollo, correspondiente a otro conjunto de servidores y servicios, se implementa en el entorno de preproducción para realizar las pruebas finales antes publicarla en la nueva intranet.

En la figura 2.14 se pueden observar las tres intranets alojadas en el servidor web. Tal y como se menciona anteriormente, parte de la información sensible de la empresa ha sido ocultada por motivos de seguridad.

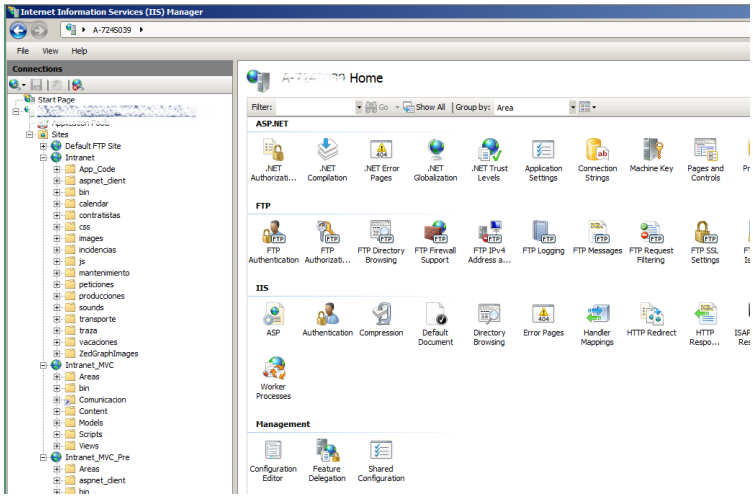


Fig. 2.14 Ejemplo de IIS en J. C. Eurosit

CAPÍTULO 3. DESARROLLO DE LA HERRAMIENTA

En este capítulo de enuncia el desarrollo de las aplicaciones que componen la elaboración de este TFC. En capítulos anteriores se ha intentado exponer un escenario de trabajo lo más claro posible que permita, en estas líneas que siguen, explicar de la forma más clara y sencilla el trabajo realizado.

La redacción de este capítulo se dividirá en varios apartados correspondientes a las diferentes áreas de programación y desarrollo realizadas.

El primer apartado versará sobre la base de datos creada y sus tablas y procedimientos almacenados correspondientes.

Seguidamente se incluye una breve explicación del proceso de creación de etiquetas de lectura/escritura en el servidor OPC asociadas a las DB del PLC a las cuales queremos tener acceso.

A continuación, en el tercer apartado, se trata el desarrollo de las aplicaciones de escritorio. Se presentan los aspectos técnicos en el desarrollo de cada una de ellas, su instalación en los puestos de trabajo en la línea de montaje y las funcionalidades que cada una de ellas aporta.

Por último, en el cuarto apartado se resume brevemente el desarrollo de las páginas web creadas para la extracción de informes de defectos así como para las pantallas de información a tiempo real distribuidas a lo largo de la línea.

Como ya se expuso en el capítulo 1, debido a la celeridad con que se habían de implementar los nuevos requerimientos acordados con del cliente, el desarrollo de este TFC se ha realizado en varias etapas. Con el objetivo de no confundir al lector, este capítulo se ha redactado teniendo en cuenta la última etapa de desarrollo del TFC. Una vez explicadas todas las funcionalidades de la herramienta completa, en el anexo final F, se indicarán las modificaciones realizadas en el proyecto de una fase de entrega a otra.

A modo de prólogo, se inicia el presente capítulo con una explicación detallada del procedimiento establecido, detallando en cada momento la intervención de cada elemento desarrollado a lo largo de este TFC.

Tal y como se puede visualizar en la figura 3.1 cuando se recibe la petición de fabricación de un asiento se inicia su fabricación en la línea de montaje. Una vez completado el ensamblado de todas sus piezas el asiento llega a la estación de Inspección de Calidad. En este puesto de trabajo se instalará la primera de nuestras aplicaciones desarrolladas, que se denominará Quality Checker.

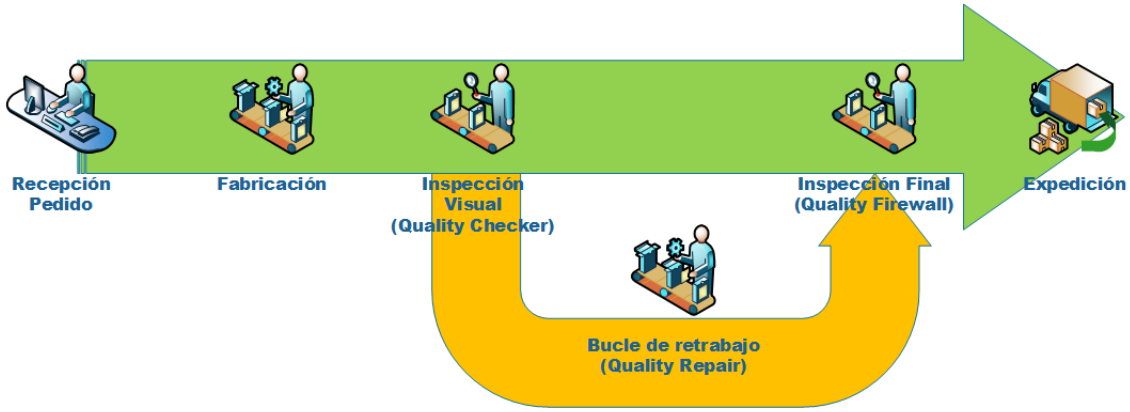


Fig. 3.1 Flujo de trabajo en la fabricación de los asientos

Esta aplicación detectará cuándo el palé se detiene en su posición, realizará una consulta al Servidor OPC que a su vez consultará las DB del PLC y devolverá los valores de la información sobre el asiento en curso a la aplicación Quality Checker.

Una vez recibida esta información, la aplicación hará una consulta a la base de datos SQL a fin de descargar el listado de posibles defectos relacionados con ese tipo de asiento. Los defectos se mostrarán organizados en varias categorías y con un orden específico (de mayor a menor importancia o recurrencia) para facilitar la experiencia del usuario y reducir el tiempo de registro de defectos.
Posteriormente al registro de los defectos, en caso de que los hubiera, el palé avanzará siguiendo el curso establecido por la línea de montaje. En el caso de que el asiento registre defectos, ya sean de calidad detectados en este puesto, o de proceso de montaje registrados en posiciones anteriores en la línea de montaje, el asiento se desviará hacía el bucle de retrabajo, del cual no saldrá hasta estar completamente reparado y en óptimas condiciones de calidad, o se dirigirá hacia el bucle de expedición.

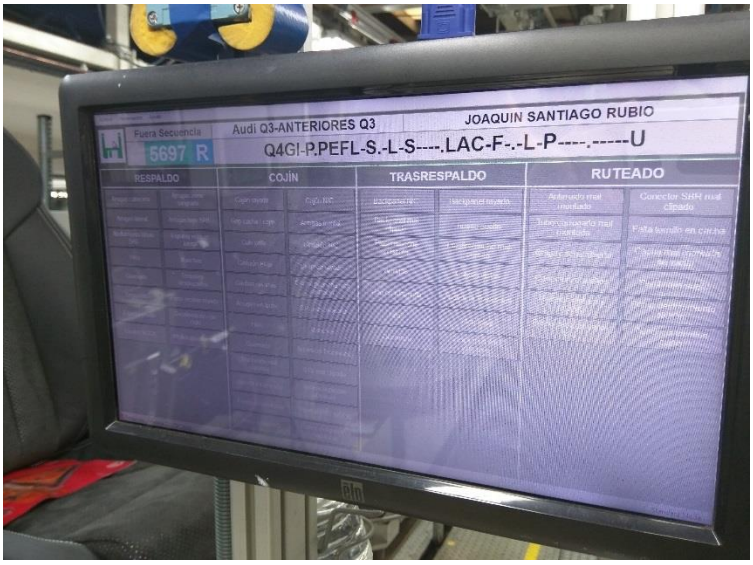


Fig. 3.2 Quality Checker

Inicialmente, la aplicación Quality Checker se programó para mantener el asiento anclado en el puesto hasta que el auditor de calidad no decidiera liberarlo. En caso de marcar, o no, y guardar defectos, la aplicación lanza una instrucción de escritura sobre un DB del PLC liberando así el asiento. La aplicación, además de liberar el asiento escribe, en una DB especial, si el asiento tiene defectos de calidad o no.
Por cuestiones de tiempos de proceso y fabricación y definición de tareas del puesto de trabajo, esta funcionalidad se deshabilitado. Actualmente la aplicación avisa al usuario cuando un asiento nuevo llega. Éste revisa el asiento y si encuentra un defecto lo registra en la pantalla táctil. Posteriormente, dispone de dos botones conectados directamente con el PLC para indicar si el asiento es correcto o erróneo y así liberarlo.

Una vez liberado el asiento se dirige hacia el bucle de retrabajo en caso de tener defectos y/o errores de fabricación o bien hacia la zona de expedición en caso de estar perfectamente ensamblado y cumpliendo los estándares de calidad establecidos.

El bucle de retrabajo es una zona de especial de la línea de producción en la que se dispone de todas las herramientas necesarias para corregir y reparar todos los posibles fallos de producción de un asiento. En el caso de la línea del Q3, dispone de varias ramas en las que alojar asientos.
Los operarios, mediante un escáner de código de barras de mano escanean la etiqueta de secuenciación del asiento que se dispongan a reparar. Está conectada con el PLC y carga los datos del asiento a reparar en un DB concreta.

Mediante una aplicación software se accede a esta DB mostrando todos los fallos de fabricación, ya sean aprietes de tornillos incorrectos, pruebas de regulaciones eléctricas erróneas u otro tipo fallos derivados del proceso de montaje. Se realiza la reparación que corresponda y la aplicación realiza de nuevo los testeos oportunos liberando así el asiento en caso de que sean correctos.

Nuestra aplicación para la detección de defecto es una versión reducida que trabaja de forma similar. La aplicación, Quality Repair, permanece en constante escucha y, cuando detecta la carga de un nuevo asiento en la DB específica antes comentada, realiza una consulta SQL en busca de defectos asociados a este asiento. Si los encuentra los muestra por pantalla en forma de botones, indicando al operario los defectos que el auditor de calidad ha detectado y que él debe de reparar. Una vez corregidos el operador debe marcar los defectos como corregidos y guardar los datos. Estos se actualizarán en la base de datos SQL indicando la fecha y hora en la que fueron corregidos.

Si bien es verdad que las funcionalidades de la aplicación Quality Repair se podían haber integrado en la aplicación de reparación de defectos de proceso y fabricación, también conocidos por defectos de Poka Yoke), se descartó en su momento por motivos de compatibilidad de software y por proceso de trabajo.

Tanto el lenguaje de programación en el que se diseñó la aplicación como sus librerías especiales para conectar con los sistemas de testeo eléctrico y funcional de varios elementos del asiento resultaban incompatibles con los utilizados en

Quality Repair. Se decidió así pues, mantener las dos aplicaciones separadas evitando que una interfiera con la otra.

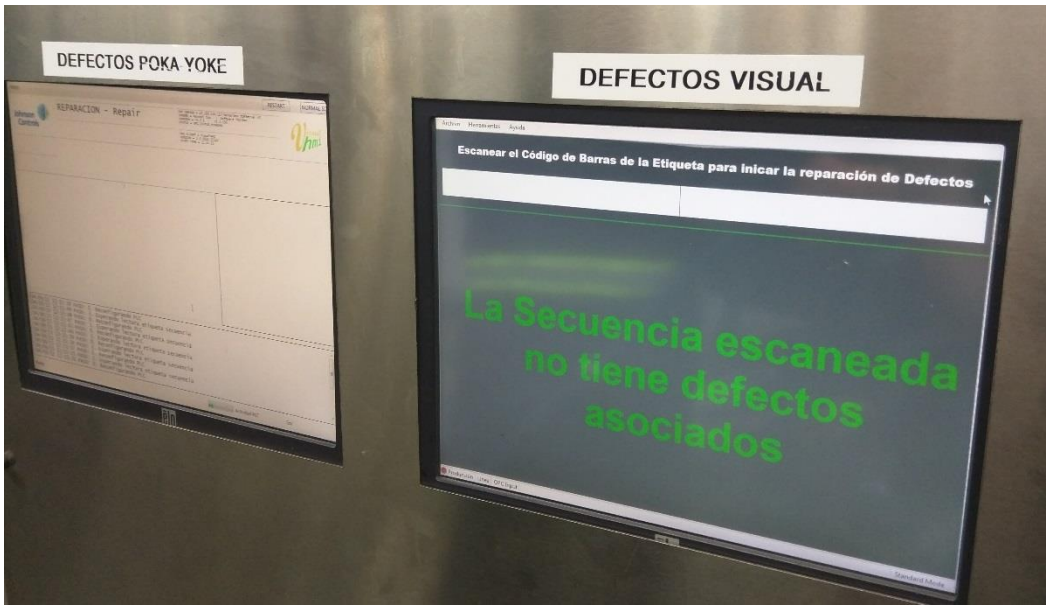


Fig. 3.3 Aplicación de corrección de defectos de POKA YOKE y Quality Repair

Finalmente el asiento, una vez corregido, se dirige hacia la zona de expedición donde, de forma secuencial, se almacenan en contenedores especiales y se envían al cliente.

En la etapa final de este TFC y, ante los buenos resultados obtenidos por este sistema, el departamento de Calidad junto con los departamentos de Producción y Procesos, deciden establecer un segundo puesto de Inspección de Calidad. A lo largo de los meses transcurridos en la elaboración de este TFC, las exigencias de calidad relacionadas con el modelo Q3 se han incrementado considerablemente. Con el objetivo de cumplir con los estándares pactados con el cliente y presentar un producto de la mayor calidad posible se decide crear una tercera aplicación, Quality Firewall, que funcionará como una mezcla de las aplicaciones Quality Checker y Quality Repair.

Cuando el asiento llega al puesto de Inspección Final, la aplicación Quality Firewall lo detecta y consulta sus datos, vía OPC. Una vez hecho esto lanza una consulta SQL al servidor de base de datos solicitando la lista de defectos a revisar para el modelo de asiento detectado. Este listado se carga en el entorno gráfico de la aplicación y permite al inspector de calidad registrar si se han detectado o no defectos. En caso de detectarlos se corrigen in situ, se marca el asiento como revisado y se expide. En caso de no tener errores se marca igualmente como revisado para indicar así al cliente que el 100% de los asientos fabricados son revisados en este puesto.

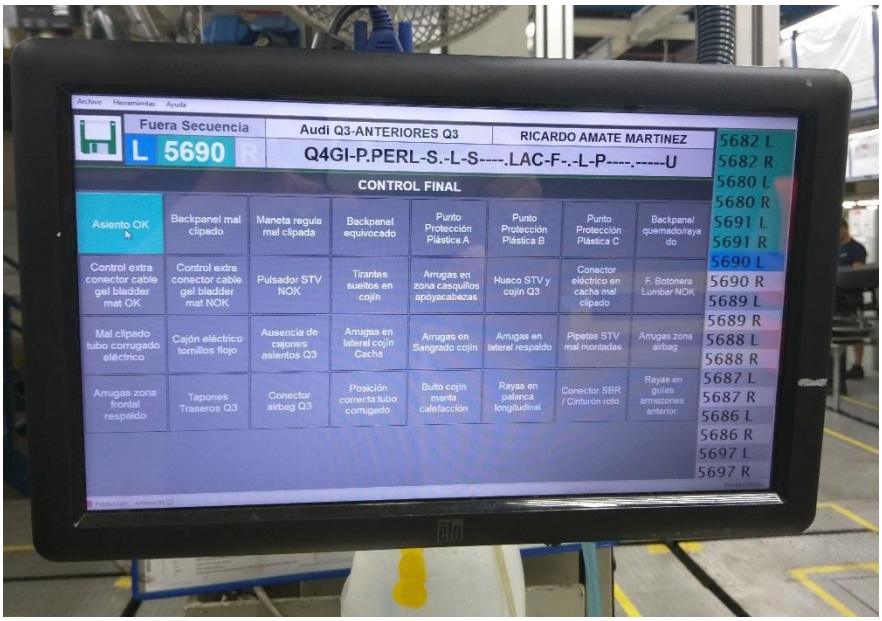


Fig. 3.4 Quality Firewall

3.1. Base de datos

En el momento de desarrollar la base de datos se han tenido en cuenta todas las especificaciones y requerimientos técnicos necesarios para el desarrollo de las aplicaciones de escritorio así como para nuevas funcionalidades.

Tanto en la creación de la base de datos, como en el desarrollo de las aplicaciones de escritorio se han intentado seguir buenas prácticas de diseño. De esta forma no solo se ha buscado cumplir con las especificaciones técnicas actuales del desarrollo de este TFC sino además convertirla en una herramienta escalable y de fácil adaptación a los cambios que podrían acontecer en un futuro. En las siguientes líneas explicaremos con detalle cómo se han diseñado las tablas y relaciones entre ellas y que campos de datos se han creado para, de una forma sencilla, actualizar tipos de defectos, categorizarlos, ordenarlos y demás opciones variables.

De nuevo, con la intención de centrarse en el desarrollo de las aplicaciones de este TFC en este apartado se expondrán las tablas diseñadas y los procedimientos almacenados creados para tal fin. En el anexo B de esta memoria se analizará de forma detallada el proceso de creación de bases de datos, tablas y procedimientos almacenados.

3.1.1. Tablas

Tras el estudio de todas las funciones requeridas para el desarrollo de este TFC se han definido las siguientes tablas de datos:

- **Defecto:** Es la tabla principal en la que se almacenarán todos los defectos detectados en un asiento. Se generará una línea por cada defecto introducido en el que se almacenará el tipo de defecto registrado, los

datos referentes al asiento como su broadcast code, número de secuencia, lbhd, etc. Por último, cada registro almacenará la fecha en la que fue detectado y corregido y el identificador del auditor de calidad que lo registró.

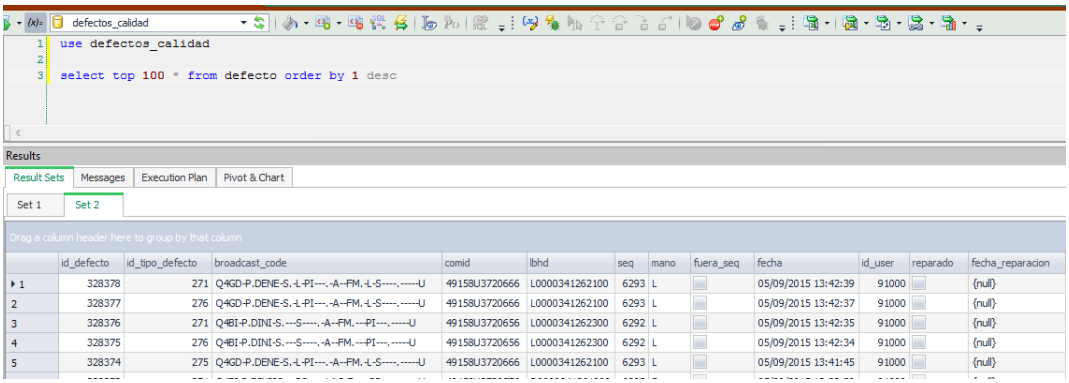


Fig. 3.5 Consulta de los últimos 100 defectos de asientos registrados

- **Ma_familia:** En esta tabla se establece la relación de categorías en las que organizar los tipos de defectos según el puesto de trabajo.
- **Ma_linea:** Almacena las líneas de producción de la planta.
- **Ma_modelo:** Registro de todos los tipos de modelos asociados a su línea de producción.
- **Ma_puesto_trabajo:** Definición de los puestos de trabajo en los que se implementará la solución software desarrollada.
- **Ma_tipo_defecto:** Esta tabla almacena el listado completo de todos los tipos de defectos por puesto de trabajo, modelo y familia. Dispone de un campo descripción, reducido para ajustarse al tamaño de los botones en las aplicaciones de escritorio y un campo más extenso en el que se almacena la descripción más detallada para los informes Excel. Además, mediante el campo orden_visual, establecemos la posición de cada defecto dentro de cada categoría de organización de defectos.
- **Usuario:** Listado de los auditores de calidad responsables de la detección de defectos.
- **Usuario_puesto:** Relación de usuarios habilitados para trabajar en cada puesto de trabajo.

En el modelo UML que se muestra a continuación en la figura 3.6 se puede observar el diseño completo de la base de datos. La definición de sus tablas, sus columnas y la relación entra cada una de ellas está presente en este esquema visual.

Como se puede observar, la mayoría de las tablas maestras, identificadas con el prefijo ‘ma’, disponen de un campo habilitado. Mediante este campo daremos de alta o de baja, usuarios, tipos de defectos, líneas de producción, modelos y puestos de trabajo según la evolución de la aplicación y la planta de montaje sin necesidad de borrar registros. Esto permite un gran escalabilidad de la aplicación de forma rápida y sencilla.

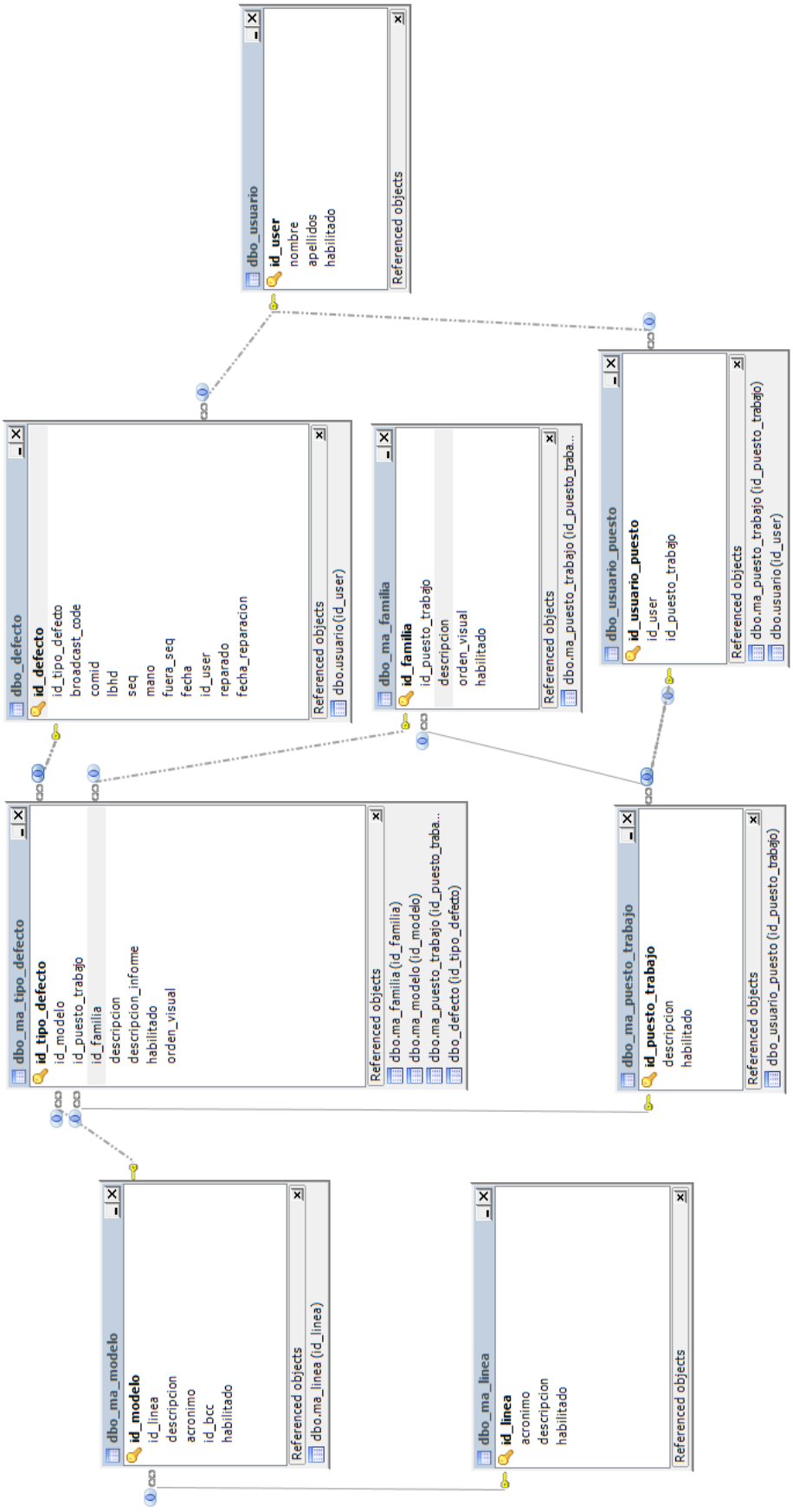


Fig. 3.6 Diagrama UML de la base de datos

3.1.2. Procedimientos almacenados

Los procedimientos creados a lo largo de este TFC se encargan de realizar diferentes tipos de consultas a la base de datos. Filtran, manipulan y devuelven datos según la funcionalidad para la que se hayan programado. De esta forma podríamos agruparlos en tres categorías y explicar, brevemente, su funcionalidad.

Como se puede observar, todos ellos tienen un tipo de nomenclatura similar. Por convenciones de código propias se decidió que todos los procedimientos almacenados deben definirse con el prefijo 'sp' inicial seguido de un código de letras que indiquen el tipo de consulta SQL a realizar y seguido de una breve explicación de su funcionalidad. Así pues, el procedimiento sp_c_asiento_lbhd es un procedimiento de consulta 'c' de asientos según lbhd, o lo que es lo mismo, pasándole por parámetro un lbhd en concreto la consulta lo buscará en la base de datos y devolverá los defectos de ese asiento si es que los tiene.

Los procedimientos cuyo funcionamiento sea realizar consultas de datos se identificarán con una 'c', como se ha visto. Si estas consultas están destinadas a funcionalidades de tipo web la nomenclatura seria 'cw'. En el desarrollo de este TFC se desarrollaran también procedimientos de modificaciones de datos 'm'. Los procedimientos para añadir datos 'a' o borrarlos 'b' no se han definido en este desarrollo. Con el objetivo de asegurar los datos registrados por la aplicación no se permite la opción de borrarlos directamente desde la aplicación. Cualquier error en el registro de un defecto es comunicado al departamento de Calidad y este, tras su análisis, realiza una petición del departamento de IT para su modificación o eliminación. En cuanto a los procedimientos para añadir datos no se han implementado. Como una excepción, las consultas de inserción de datos se han incluido en el código de las aplicaciones de escritorio como excepción al patrón de diseño antes mencionado respecto a evitar este tipo de prácticas. En el apartado dedicado al desarrollo de las aplicaciones de escritorio se explicará el motivo de por qué esta excepción.

Los procedimientos almacenados diseñados en este TFC se pueden dividir en tres grupos según las aplicaciones que los utilizan y sus funcionalidades.

En el primer gran grupo se encuentran los procedimientos dedicados a la obtención de datos de tablas para el funcionamiento de las aplicaciones de escritorio. A continuación se nombraran los procedimientos diseñados y una breve explicación de cada uno de ellos:

- **Sp_c_asiento_lbhd:** Utilizado en la aplicación Quality Repair. Al pasarle el label header (lbhd) o identificador único de un asiento, éste devuelve toda su información en el caso de encontrar defectos no reparados asociados a él. Si no los encuentra, la aplicación lo informa y espera por el siguiente asiento a revisar. En caso contrario, realiza una llamada al procedimiento sp_c_defectos_lbhd.
- **Sp_c_defectos_lbhd**: Utilizado en la aplicación Quality Repair. Este procedimiento es ejecutado después de detectar un asiento con defectos asociados en el bucle de reparación. Al ejecutarlo devuelve la lista de

defectos no reparados del asiento en cuestión para mostrarlos por pantalla en la aplicación Quality Repair.

- **Sp_c_defectos_modelo**: Devuelve el listado completo y ordenado de todos los defectos activos para un modelo y una familia en concreto.
- **Sp_c_familias_modelo**: Devuelve el listado de familias en las que organizar los diferentes tipos de defectos de un modelo en concreto para un puesto de trabajo.
- **Sp_c_lineas_activas:** Devuelve el listado de líneas de producción activas en la planta para configurar parámetros de las aplicaciones de escritorio.
- **Sp_c_modelos_activos**: Devuelve una lista de modelos activos por línea de trabajo útil para configurar parámetros de las aplicaciones de escritorio.
- **Sp_c_usuarios:** Devuelve la lista de usuarios habilitados para trabajar en un puesto de trabajo en concreto de una línea de montaje especifica.
- **Sp_m_repara_defecto**: Utilizado en la aplicación Quality Repair, modifica el registro de un defecto indicando que ha sido reparado y la fecha de esta acción.

El segundo grupo de procedimientos almacenados se encuentran aquellos diseñados para la elaboración del informe Excel vía web. El procedimiento **sp_c_defectos**, recibe por parámetro un fecha inicio y una fecha fin y, opcionalmente la línea de montaje y/o el modelo. Según estos parámetros, devuelve todos los defectos detectados en ese intervalo de fechas. Con esta información, la página web que lo llama podrá generar el informe Excel a exportar. En cuanto al procedimiento **sp_c_numero_defectos,** devuelve el número de defectos encontrados según los parámetros que antes se ha comentado para mostrarlos por pantalla en la misma página web.

En el siguiente grupo, los procedimientos almacenados **sp_cw_defectos_turno**, **sp_cw_top5_hora**, **sp_cw_top5_hora_vs_turno**, **sp_cw_top5_turno** se definen con el objetivo de obtener ciertos datos necesarios para la creación de los informes en tiempo real de defectos detectados y corregidos. A grandes rasgos, estos procedimientos realizan una serie de consultas a la base de datos para devolver la relación de defectos detectados por cada tipo a lo largo de un turno de producción, así como los 5 defectos más detectados por turno y hora de producción y su comparativa entre ellos. Gracias a estos procedimientos se elaboraran una serie de gráficos y tablas de resultados que se detallaran en el último apartado de este capítulo.

3.2. Etiquetas de lectura OPC

Una vez creada la base de datos y los diferentes procedimiento almacenados para realizar las inserciones y consultas de datos, llega el momento desarrollar los métodos requeridos para obtener la información del asiento de forma automatizada.

Para ello, se definirán una serie de etiquetas o identificadores únicos en el servidor OPC que estarán asociados a las DBs del PLC a los que se quiera acceder.

Mediante el programa TOP Server, instalado en el servidor OPC, se generará una estructura de datos asociada a cada puesto de trabajo. Esta estructura, tal y como se muestra en la figura 3.7, se compone de canales, componentes, grupo de etiquetas y etiquetas.

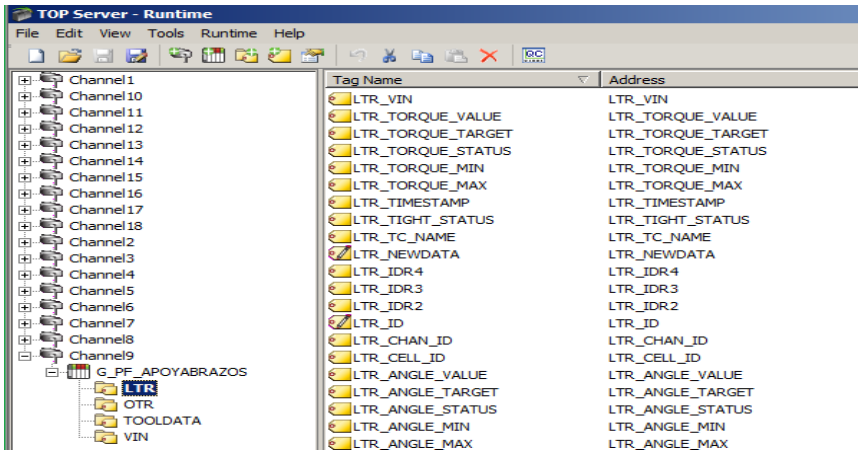


Fig. 3.7 Estructura de datos OPC

Para simplificar este apartado tan solo se introducirá el concepto de etiqueta OPC. En el anexo C de esta memoria se encuentra una explicación detallada de cómo crear una estructura OPC compuesta por un canal, al cual se asociaran los diferentes dispositivos a los que se quiera conectar. Sobre estos dispositivos se establecerán los grupos de etiquetas y las etiquetas deseadas.

Para el desarrollo de este TFC se crearán dos canales, uno para el PLC de la línea de producción del Ibiza y otro para la línea de producción del Q3. Si bien es verdad que ambos PLC son idénticos y se podría gestionar con el mismo canal, se decide esta separación para evitar que los cambios en una línea puedan afectar a la configuración de la otra.

Dentro de cada canal se creará un dispositivo correspondiente a cada PLC y, para cada dispositivo, se crearan tres grupos de etiquetas, correspondientes a los tres puestos de trabajo, inspección visual, reparación e inspección final. Cada grupo contendrá las etiquetas correspondientes a los campos de información que se necesitan obtener para el diseño y funcionamiento de las aplicaciones de escritorio.

Tag Name	Address	Data Type	Scan Rate	Scaling
VIN	DB214.STRING102.14	String	100	None
SubConjunto	DB214.C127	Char	100	None
SlotContenedor	DB214.I130	Short	100	None
Ship	DB214.C125	Char	100	None
Secuencia	DB214.I118	Short	100	None
Orden	DB214.DI120	Long	100	None
Linea	DB214.C124	Char	100	None
LBHD	DB214.STRING14.20	String	100	None
Lado	DB214.C126	Char	100	None
Contenedor	DB214.I128	Short	100	None
BroadcastID	DB214.DI98	Long	100	None
BroadcastCode	DB214.STRING36.60	String	100	None
BCSAB	DB214.STRING190.20	String	100	None
BCCover	DB214.STRING212.28	String	100	None

Fig. 3.8 Etiquetas OPC. Puesto de Inspección Visual del modelo Q3

3.3. Desarrollo de aplicaciones de escritorio

Una vez definida la base de datos y configuradas las etiquetas OPC que permiten el acceso a los datos del PLC, es el momento de desarrollar las aplicaciones de escritorio para los terminales de los puestos de montaje.

Como ya se ha comentado anteriormente se han desarrollado tres aplicaciones cada una con una funcionalidad pero que comparte muchas características y funciones.

En este apartado se explican estas funcionalidades en común como son el acceso a Base de Datos y las etiquetas OPC, los ficheros de configuración de las aplicaciones y su instalación en los terminales.

Por último se tratarán cada una de las aplicaciones desarrolladas explicando su funcionamiento y peculiaridades.

3.3.1 Funcionalidades comunes de las aplicaciones

Las tres aplicaciones desarrolladas a lo largo de este TFC, como ya se ha dicho, comparten un gran número de funcionalidades y métodos. A lo largo del proceso de creación de código se ha intentado reutilizar al máximo posible las funciones y rutinas dotándolas de un carácter genérico que permitiera así el uso por parte de los tres proyectos de programación.

Tanto el aspecto visual de los formularios como las funciones básicas para alimentar los campos de datos se han programado teniendo siempre en mente el funcionamiento de las tres aplicaciones. De igual forma, las tres aplicaciones comparten los mismos métodos de conexión a BBDD y de lectura y escritura en las etiquetas OPC así como el fichero de configuración en su instalación.

J. C. Eurosit dispone de un gran número de terminales instalados en la línea de montaje. Estos terminales son una herramienta indispensable para los operarios así como un elemento fundamental en algunos puestos de trabajo, en los cuales, realizan operaciones de testeo o monitorización de procesos que influyen directamente sobre el proceso de montaje. Es por eso que estos terminales, en algunos casos críticos por las funcionalidades que realizan, deben ser fácilmente sustituibles en el caso de algún tipo mal funcionamiento por su parte.

Por este motivo en J. C. Eurosit se tiende a utilizar en las líneas de montaje el mismo modelo de terminal con capacidad suficiente para trabajar todos los programas implementados en dichos equipos. En la actualidad conviven en J. C. Eurosit tres modelos de Thin Clients, dos de ellos de características similares y un tercero en proceso de sustitución.

El Departamento de IT de J. C. Eurosit cuenta con un número de Thin Clients de backup configurados y listos para sustituir en cualquier momento un equipo que deje de funcionar en la línea de montaje. Es por eso que a todos los Thin Clients

de J. C. Eurosit se les instala una imagen de fábrica idéntica que contiene todos los programas necesarios para sustituir a cualquier equipo de la línea de montaje

Tal y como se puede observar en la figura 3.9 dentro de la ruta C:\Program Files\Johnson Controls Eurosit se encuentran tres carpetas correspondientes a cada una de las aplicaciones. En su interior se encuentra el fichero ejecutable de la aplicación (*.exe), el fichero de configuración (*.config) y la librería XmlDANet.dll que permite establecer conexión con el servidor OPC y acceder a las etiquetas OPC configuradas para cada puesto de trabajo.

Adicionalmente, el programa Quality Firewall, dispone de un fichero de texto en el que se almacenan las secuencias comprendidas entre el puesto de Inspección Visual de Calidad y el puesto de Inspección Final. La utilidad de fichero se comentará más adelante cuando se explique con detalle el funcionamiento de esta aplicación.

Para el funcionamiento de las aplicaciones desarrolladas, además de los ficheros comentados, es necesario que el terminal tenga instalado el Framework de Microsoft .NET, en nuestro caso la versión 4.5.2 con el que han sido desarrolladas estas aplicaciones.

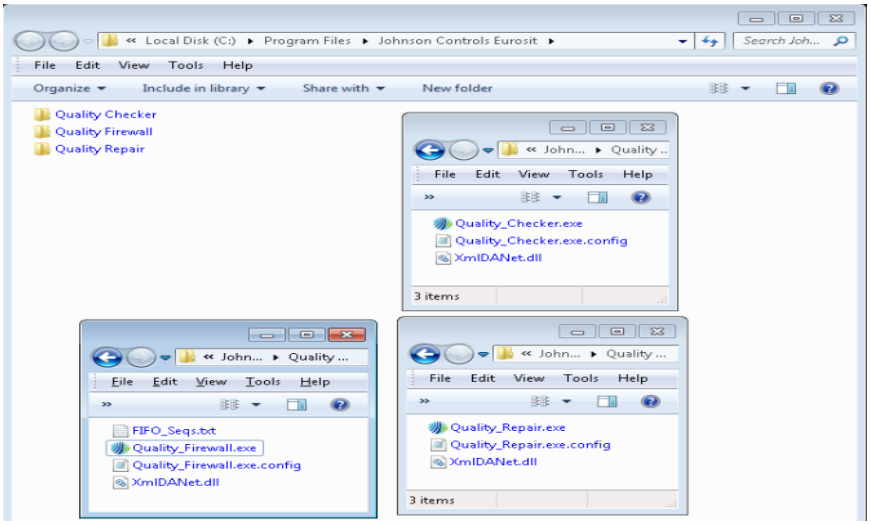


Fig. 3.9 Ficheros necesarios para la ejecución de los programas desarrollados

En el momento de desarrollar las aplicaciones se estableció que estas debían de funcionar tanto para la línea de producción del modelo Ibiza como la del modelo Q3, así como también debían de estar preparadas para adaptarse a la implantación de nuevas líneas de producción en la planta. Es por este motivo que se decidió extraer del código todas las variables de configuración referentes al puesto de trabajo así como las cadenas de conexión a los servidores OPC, SQL y dispositivos OPC de las diferentes líneas implementadas. En el anexo D incluido en esta memoria se puede observar un ejemplo de este fichero de configuración y la explicación de cada uno de sus parámetros.

Los terminales situados en los puestos de montaje son ordenadores conectados a la red de Johnson Controls de fácil acceso para cualquier trabajador, interno y externo, de la planta. Si bien están conectados en una VLAN propia aislada del resto de tráfico de la red local y sin acceso al exterior, las aplicaciones que se programan para estos terminales deben de reforzar su seguridad y evitar la manipulación de personal no autorizado.

Las aplicaciones desarrolladas arrancan al encender los terminales y no se pueden cerrar si no se activa el modo Administrador mediante la contraseña antes mencionada. En el caso de que, por cualquier circunstancia, se pudiera visualizar el fichero de configuración, los datos sensibles de las cadenas de conexión a BBDD y servidores OPC están cifrados con un software propio.

Este módulo está integrado en las aplicaciones desarrolladas para así poder cifrar las cadenas de conexión y trabajar con los servidores de bases de datos y OPC. La aplicación utiliza un algoritmo de cifrado MD5 de 128 bits. En la figura 3.10 se muestra un ejemplo de la aplicación de cifrado y descifrado de desarrollo propio.

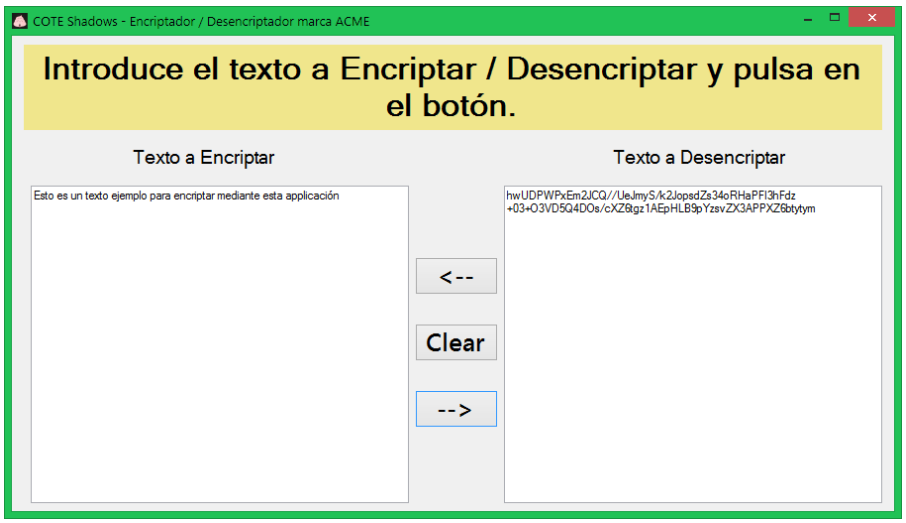


Fig. 3.10 Aplicación de desarrollo propio para el cifrado de cadenas de conexión

Respecto a las cadenas de conexión utilizadas en el desarrollo de este proyecto, a continuación se muestra un ejemplo de su sintaxis:

Ejemplo de cadena de conexión a BBDD SQL:

```
DataSource=<ServidorBBDD>;Initial Catalog=defectos_calidad;
Persist Security Info=True; User ID=<Usuario>;
Password=<Contraseña>
```

Ejemplo de cadena de conexión a Servidor OPC:

```
x.x.x.x/SWToolbox.TOPServer.V5
```

3.3.2 Quality Checker

Quality Checker es la aplicación diseñada para registrar todos los defectos detectados por el auditor de calidad en el puesto de Inspección Visual de la línea de montaje.

El flujo de trabajo de esta aplicación es muy sencillo, tal y como se puede ver en la figura 3.11.

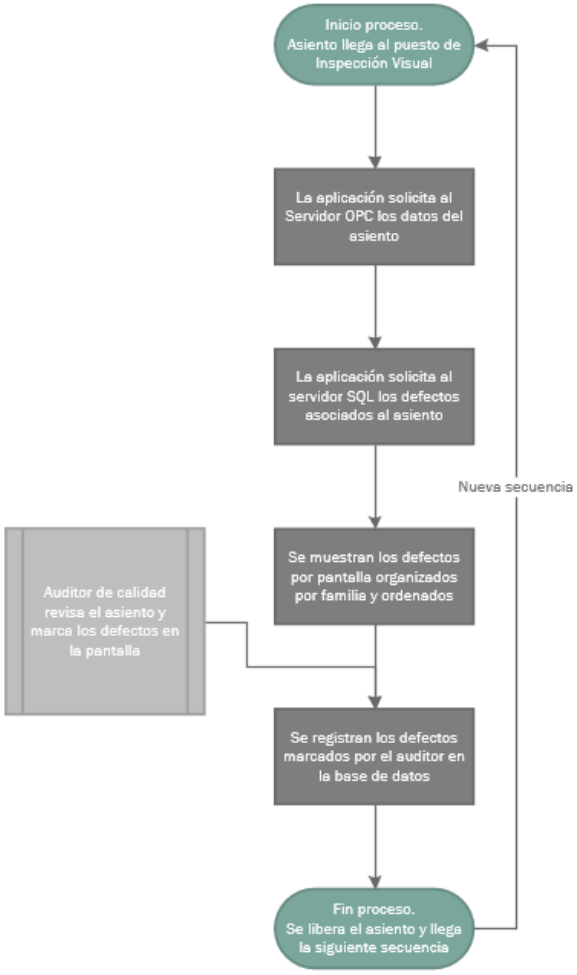


Fig. 3.11 Flujo de trabajo de Quality Checker

Como se puede observar en la figura del flujo de trabajo, la aplicación espera por la llegada de un asiento al puesto de montaje. En el momento en el que éste se detiene, el PLC actualiza la DB correspondiente con la información del asiento. La aplicación a su vez detecta el cambio de secuencia y obtiene los datos de las etiquetas OPC correspondientes. Con esta información, se hace una solicitud al procedimiento almacenado de base de datos correspondiente para obtener las familias y tipos de defectos relativos a ese modelo de asiento. De esta forma, carga el formulario de defecto agrupados por familias y ordenados según frecuencia y espera la interacción del auditor de calidad. Éste marcará los defectos de calidad en el caso de que los tenga y liberará el asiento reiniciándose el ciclo con la siguiente secuencia.

Si bien éste es el modo de funcionamiento diseñado en un principio, tras un estudio de tiempos y ergonomía, con el objetivo de minimizar el tiempo de proceso de la estación de trabajo y los movimientos repetitivos, se modificará el proceso hacia la versión actual.

Por normal general no todos los asientos tienen defectos de calidad, es por eso que, tras los estudios realizados por los departamento de Calidad y Procesos se decide separar el funcionamiento de la aplicación del flujo de trabajo de la línea de montaje. De esta forma la aplicación no actualiza los datos de secuencia en el momento en el que ésta se detiene en el puesto de montaje. Tal y como se muestra en la figuras 3.12 y 3.13, el fondo del formulario de defectos cambia a un color rojo sin modificar los valores de secuencia ni el listado de defectos asociados a ésta.

Fig. 3.12 Información de secuencia no actualizada en Quality Checker

Fig. 3.13 Información de secuencia actualizada en Quality Checker

El fondo de color rojo indica al auditor de calidad que los datos que se muestran por pantalla no corresponden al asiento que se encuentra actualmente en el puesto de trabajo, avisándole para que evite asociar defectos a asientos que no le corresponden.

Para iniciar el proceso recibiendo los datos del asiento en curso, el auditor debe pulsar sobre el botón de guardar salvando los datos del último asiento actualizado y refrescando el formulario con los datos del asiento actual. En el caso de no haber marcado ningún tipo de defecto para el asiento anterior tan solo se ejecutaría la siguiente operación.

Así, el auditor de calidad solo ejecuta la operación de registro de defectos en el momento en el que detecta alguno en un asiento. En ese momento refrescará la información de la pantalla y guardará los defectos localizados. Mientras tanto, la aplicación no interactuará con la línea de montaje evitando de esta forma el movimiento repetitivo de liberar asientos cuando no se detecten defectos.

Además se dispone también de un modo de trabajo adicional. En la primera fase de desarrollo se implementó el modo Escáner para aportar una solución funcional en un corto espacio de tiempo mientras se implementaba la actual forma de trabajar. En el llamado modo Escáner es el propio auditor el que inicia el proceso escaneando el código QR de la etiqueta de secuenciación.

El escáner de mano pasa a la aplicación la información contenida en la etiqueta y ésta actualiza los campos del formulario, a su vez realiza la petición a la base de datos sobre los defectos asociados al asiento recién escaneado. De esta forma el proceso continúa de forma idéntica a los explicados anteriormente.

A pesar de que el modo de trabajo por defecto es el que emplea la conexión con el servidor OPC, se mantiene en el código la funcionalidad del modo Escáner permitiendo su cambio tanto desde el fichero de configuración como desde la aplicación. En este último caso con permisos de administración.

Finalmente, una vez el auditor de calidad marque todos los defectos asociados al asiento, el programa creará una conexión a la base de datos para cargar de forma masiva un registro por cada defecto marcado. En este caso, se ha decidido incluir una instrucción de inserción de datos SQL, en vez del habitual uso de procedimientos almacenados, para evitar tener que hacer una llamada por cada registro a insertar.

Mediante la instrucción SQLBulkCopy del Framework.NET se insertan todos los registros de una sola vez. Para ello se crea una tabla en VB.NET con la misma estructura de campos que la tabla SQL en la que se quiere hacer la inserción. Una vez definida se inserta una línea a esa tabla con los datos de los registros de cada defecto. Finalmente, se establece una conexión con la base de datos, mapeando las columnas de la tabla de la aplicación con la tabla SQL y se ejecuta la instrucción para realizar la inserción.

Quality Checker dispone además de funcionalidades cerrar y abrir una nueva sesión de trabajo, para permitir el cambio de usuario. Tanto el cambio de entorno

de trabajo, de producción a desarrollo, y el cierre de la aplicación solo se permiten accediendo al modo administrador.

Estas funcionalidades también las se encuentran presentes en las otras dos aplicaciones desarrolladas.

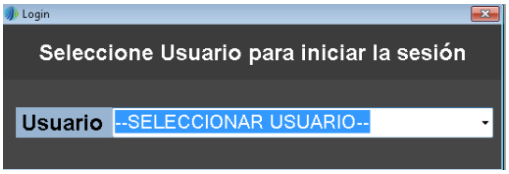


Fig. 3.14 Formulario para el inicio de sesión de usuario

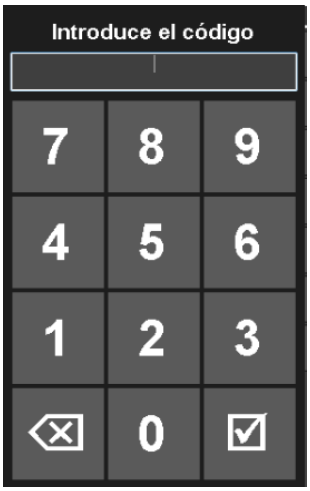


Fig. 3.15 Formulario para introducir el código y habilitar el modo administrador

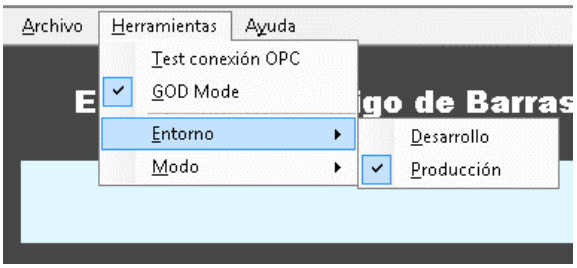


Fig. 3.16 Barra de herramientas para el cambio de entornos, habilitar el modo administrador y acceder al testeo de conexión OPC

Finalmente, la última característica en común de las tres aplicaciones, corresponde a un formulario para el testeo de la conexión OPC útil en el momento de desarrollo de estas aplicaciones.

Fig. 3.17 Formulario de testeo de conexión OPC

3.3.3 Quality Repair

De las tres aplicaciones desarrolladas Quality Repair es la más sencilla, tanto a nivel de programación como a nivel de operativa.
Diseñada utilizando como proyecto base el creado para la aplicación Quality Repair, hereda gran parte de sus formularios, funciones y diseño gráfico.

Al igual que su aplicación hermana, Quality Repair puede funcionar en los modos Escáner/OPC apuntando a los entornos de Desarrollo y Producción. También dispone de un modo Administrador mediante el cual podemos modificar los entornos configurados por defecto y cerrar la aplicación.

A nivel funcional, Quality Repair muestra un flujo de trabajo muy sencillo. El operario del puesto de retrabajo escanea la etiqueta del asiento que se dispone a verificar y reparar y se inicia el proceso.

En el caso de estar configurado en modo Escáner, la aplicación lee los datos del asiento del código QR y extrae el lbhd de la misma forma que se hace en la aplicación Quality Checker. Para esta casuística, el escáner de mano debe de estar conectado al terminal mediante su conexión USB.

Si por el contrario, es el modo OPC el que está operativo, en el momento de escanear el código QR se lanza una instrucción al PLC para cargar los datos del asiento a inspeccionar en la DB específica del bucle de retrabajo. En este caso el escáner de mano está conectado directamente al PLC y éste es el que, ejecutando una de sus rutinas programadas recibe el identificador del asiento, lo busca en su buffer de memoria y lo carga en la DB antes comentada.
La aplicación, por su parte, detecta los datos cargados del asiento y obtiene el lbhd de la etiqueta OPC correspondiente.

Una vez obtenido el lbhd, ya sea mediante un modo u otro, la aplicación ejecuta una consulta SQL invocando el procedimiento almacenado sp_c_asiento_lbhd, el cual devuelve información sobre el asiento si encuentra defectos no reparados.

En este caso, la aplicación ejecuta una nueva consulta SQL solicitando los defectos registrados y no reparados de este asiento. Con la respuesta por parte del procedimiento almacenado correspondiente se genera, en el interfaz gráfico, un botón por cada defecto. Hecho esto el operario, tras reparar los asientos, los marcará y guardará la información, momento en el cual un último procedimiento almacenado se ejecutará modificando estos registros indicando que se han reparado y la fecha en la que se han hecho.

Fig. 3.18 Quality Repair mostrando errores asociados a un asiento

En el caso de no devolver defectos asociados al asiento se mostrará un mensaje por pantalla, tal y como se puede observar en la figura 3.19, indicando que el asiento no tiene defectos asociados.

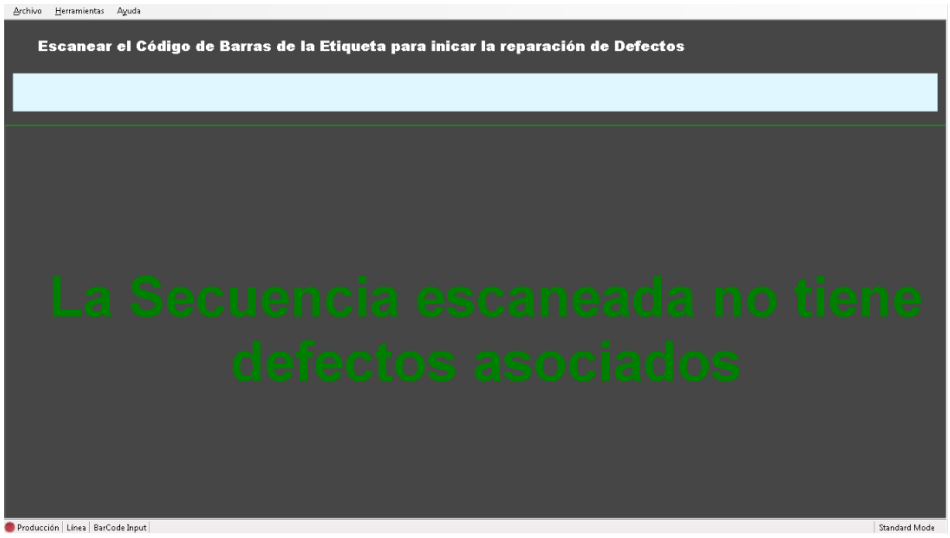


Fig. 3.19 Mensaje para un asiento sin errores asociados

Tras esto la aplicación esperará por el próximo asiento a analizar

En la siguiente figura se muestra un esquema del flujo de trabajo de esta aplicación.

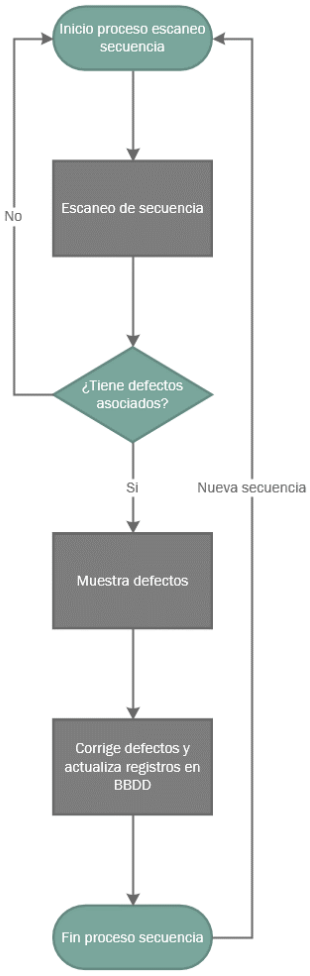


Fig. 3.20 Proceso Quality Repair

3.3.4 Quality Firewall

La última de las tres aplicaciones a desarrollar en este TFC. Quality Firewall nace en una última etapa de desarrollo con el objetivo, por parte de J. C. Eurosit, de proporcionar un asiento de calidad óptima y cumplir así su objetivo de 0 defectos en producto acabado.

Esta aplicación se define, en un inicio, como un clon de la aplicación Quality Checker, con la diferencia de disponer de un listado de tipos de defectos a revisar diferente.

Para la primera versión de esta aplicación se definen unas nuevas etiquetas OPC que apuntan a las DBs correspondientes a la estación anterior a la expedición del asiento y se adapta la base de datos para manejar un listado de tipos de defectos diferente al ya establecido.

Sin embargo, tras unos días en uso, se requieren una serie de modificaciones por parte de los departamentos de Calidad, Procesos y Producción que hacen evolucionar la herramienta hasta lo que es ahora.

Por una cuestión de tiempos de proceso, se decide que la aplicación debe listar todos los asientos que se encuentren entre el puesto de montaje de Inspección Visual y el puesto de Expedición. Esta lista de asientos permanecerá en un extremo del interfaz gráfico, de forma que permita al operador marcar cualquiera de los asientos de la lista y realizar el proceso de Inspección Final sin la necesidad de que éste se encuentre en el puesto indicado.

Este cambio en la funcionalidad permite la movilidad del operador a lo largo del bucle de rechazo y de la banda de expedición revisando asientos que ya estén listos para ello sin la necesidad de esperar a que lleguen al puesto.
Ya que los asientos se han de almacenar en el contenedor de expedición de una forma específica para que concuerden con la secuenciación del cliente, puede darse el caso de que se detenga la línea por la reparación de algún asiento defectuoso. En este caso ningún asiento del bucle podría avanzar y, asientos que podrían pasar la revisión final, no lo harían por no llegar al puesto de montaje. Con esta modificación se permite esta forma de adelantar trabajo en momentos en los que la reparación de algún asiento bloque la expedición.

Para implementar esta funcionalidad se modifica un poco el formulario de detección de errores incluyendo el listado de secuencias ya mencionado. Este listado dispone de unas 20 secuencias, capacidad máxima del bucle de rechazo. La aplicación está conectada a las etiquetas OPC del puesto de Inspección Visual, igual que Quality Checker, e incluye cada secuencia detectada en dicha estación. Esta lista definida en el programa no es más que una cola FIFO de 20 posiciones que va desplazando las secuencias conforme son detectadas en el puesto de Inspección Visual.
Para preservar el listado de secuencias ante el posible reinicio de la aplicación o la estación de trabajo, cada vez que se modifica la lista se sobrescribe el fichero FIFO_Log.txt en el que se almacena los datos de las secuencias registradas.
El siguiente texto es un ejemplo de los valores almacenados en la lista de secuencias.

```
6288;R;Q4GI-P.PEST2S.-LQS----.-AC-F-.-LQPI---.S----U;S;3;49158U3710392 ;R0000341259000;True
6287;L;Q4GI-P.DENE-S.-L-S----.-AC-F-.-L-P----.-----U;S;3;49158U3720267 ;L0000341259100;True
6287;R;Q4GI-P.DENE-S.-L-S----.-AC-F-.-L-P----.-----U;S;3;49158U3720267 ;R0000341259200;True
6286;L;Q4BI-P.DINI-S.---S----.LA--FM.---PI---.-----U;S;3;49158U3720655 ;L0000341259300;True
6286;R;Q4BI-P.DINI-S.---S----.LA--FM.---PI---.-----U;S;3;49158U3720655 ;R0000341259400;True
6297;R;Q4BI-P.DINI-S.---S----.LAC-F-.---PI---.-----U;S;3;49158U3720688 ;R0000341261400;True
6296;R;Q4BD-P.NARB2S.ELQP----.LAC-F-.ELQS----.-----T;S;3;49158U3720690 ;R0000341261600;True
6295;L;Q4BI-P.DINI2S.--QS----.LAC-F-.--QP----.-----U;S;3;49158U3720576 ;L0000341261700;True
6295;R;Q4BI-P.DINI2S.--QS----.LAC-F-.--QP----.-----U;S;3;49158U3720576 ;R0000341261800;True
6293;L;Q4GD-P.DENE-S.-L-PI---.-A--FM.-L-S----.-----U;S;3;49158U3720666 ;L0000341262100;False
6292;L;Q4BI-P.DINI-S.---S----.-A--FM.---PI---.-----U;S;3;49158U3720656 ;L0000341262300;False
6303;R;Q4GI-P.DENE-S.-L-S----.LAC-F-.-L-P----.-----U;S;3;49158U3710394 ;R0000341264400;False
```

En este ejemplo de secuencias se puede ver cómo se almacenan los datos de numeración de secuencia, mano, broadcast code, fuera de secuencia, línea, vin code, lbhd y una variable binaria en la que se indica si el asiento ya ha sido inspeccionado, o no.
En caso de que se vuelva a marcar un asiento ya inspeccionado la aplicación mostrará en mensaje que se puede ver en la figura 3.21.

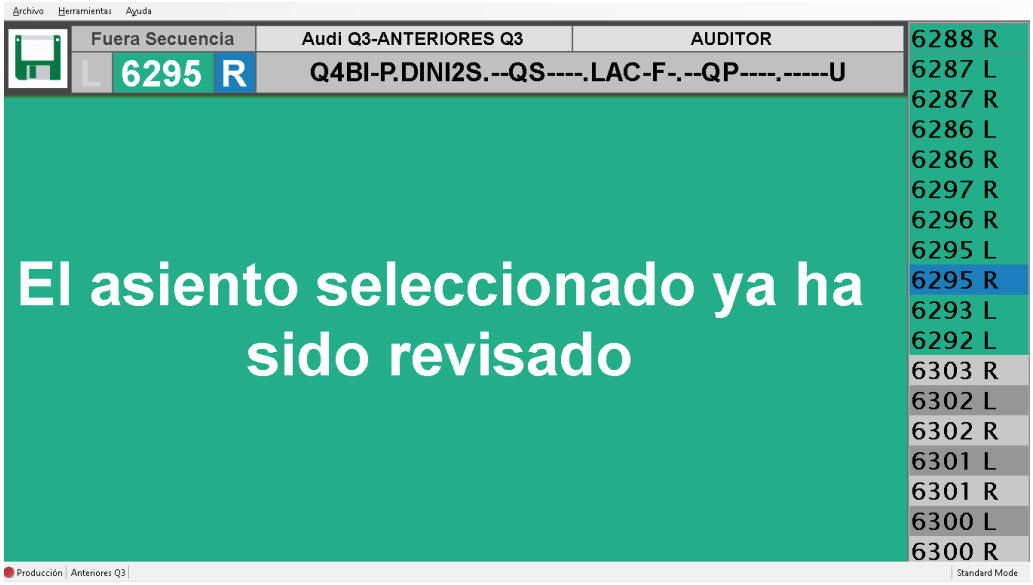


Fig. 3.21 Quality Firewall, asiento ya inspeccionado.

En caso de pulsar sobre un asiento no inspeccionado, se mostrará la pantalla de la figura 3.22.

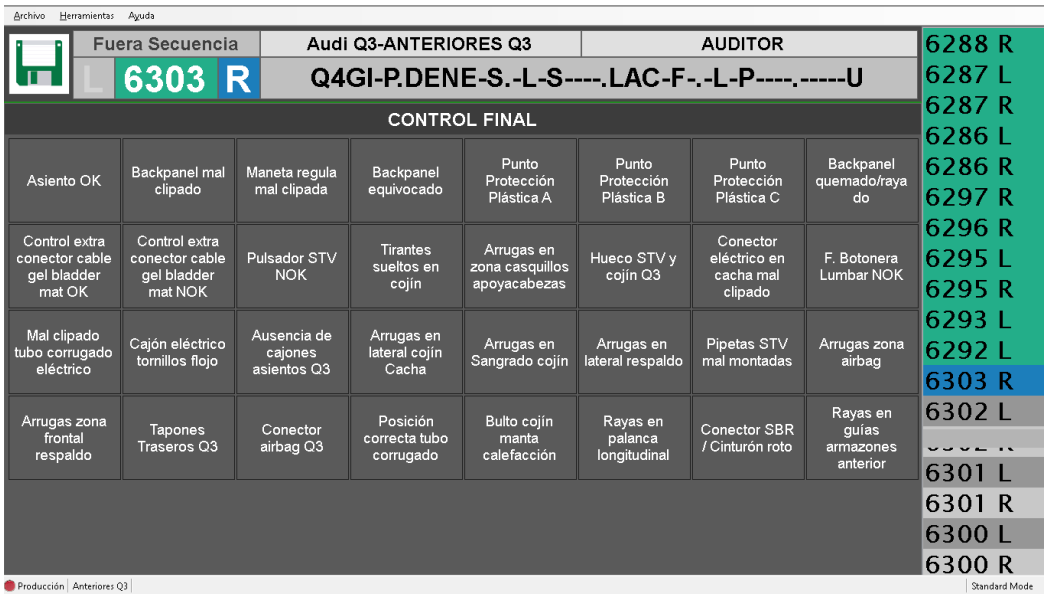


Fig. 3.22 Quality Firewall, asiento por inspeccionar

El resto del procedimiento para registrar los defectos en la base de datos permanecería igual que en la aplicación Quality Checker, tras marcar todos los defectos registrados y pulsar en el botón de guardado identificado por el icono del disquete, se ejecutará una rutina de la aplicación que creara una conexión con la tabla defectos de la base de datos e insertará los datos de forma masiva con una instrucción SQLBulkCopy.

3.4. Desarrollo de páginas web

El desarrollo de este TFC, como ya hemos repetido más de una vez, además del diseño de las aplicaciones software comentadas en el apartado anterior, tiene el objetivo de ofrecer al personal de Calidad y Producción una herramienta eficaz, rápida y robusta para actuar de forma directa sobre los defectos detectados. Para ello, se han diseñado una serie de aplicaciones web que permiten extraer información en tiempo real de todos los defectos registrados en el proceso.

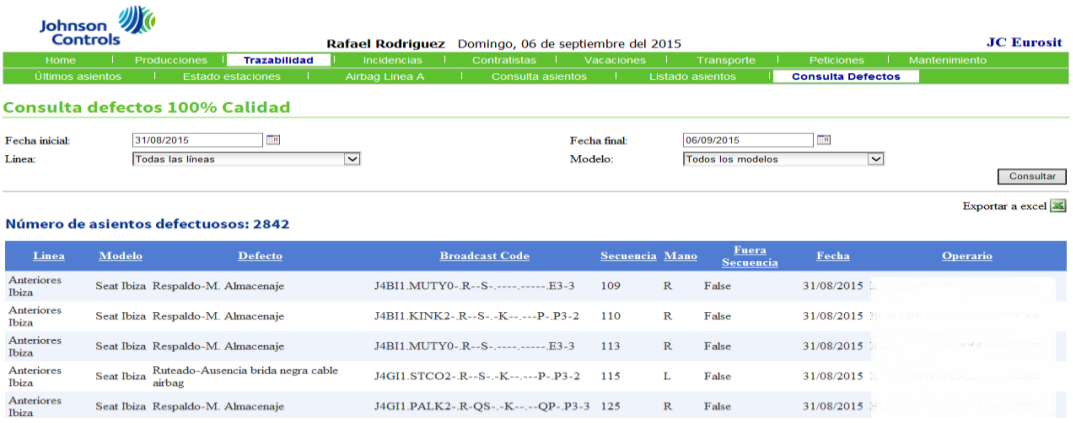
Estas aplicaciones se componen de un formulario web que permite filtrar los datos recopilados y exportarlos a un fichero Excel. Además de una serie de páginas webs que muestran gráficos y tablas de datos referentes a los defectos detectados según la hora y el turno de producción, y que van rotando de forma periódica mostrándose en unas pantallas instaladas a pie de línea de montaje.

J. C. Eurosit dispone de un entorno de intranet en el que alojar estas aplicaciones web. Como ya se comentó en apartados anteriores, esta intranet se encuentra en un proceso de migración en la cual conviven páginas programadas en ASP Web Forms y ASP.Net MVC.

Debido a la necesidad del departamento de Calidad de explotar los datos recopilados y analizarlos en un corto espacio de tiempo tras la puesta en producción del sistema de registro, las páginas web creadas en el desarrollo de este TFC se han diseñado en ASP Web Form, si bien es verdad que, aprovechando una fase de migración de aplicaciones, las pantallas de información en tiempo real implantadas a pie de línea ya han sido migradas a ASP.NET MVC.

En el anexo E de esta memoria se explican las principales diferencias entre ambos entornos de programación web. En este apartado tan solo se mostrarán los informes y las pantallas de información en tiempo real creadas.

En la figura 3.23 podemos observar el aspecto del formulario de exportación de datos. En éste, como ya hemos comentado, tras introducir una fecha inicio y fin se obtiene el número de defectos registrados que podemos exportar al informe de Excel que se muestra en la figura 3.24.



Linea	Modelo	Defecto	Broadcast Code	Secuencia	Mano	Fuera Secuencia	Fecha	Operario
Anteriores Ibiza	Seat Ibiza	Respaldo-M. Almacenaje	J4BI1.MUTY0-.R--S-.----.-----.E3-3	109	R	False	31/08/2015	
Anteriores Ibiza	Seat Ibiza	Respaldo-M. Almacenaje	J4BI1.KINK2-.R--S-.-K--.---P-.P3-2	110	R	False	31/08/2015	
Anteriores Ibiza	Seat Ibiza	Respaldo-M. Almacenaje	J4BI1.MUTY0-.R--S-.----.-----.E3-3	113	R	False	31/08/2015	
Anteriores Ibiza	Seat Ibiza	Ruteado-Ausencia brida negra cable airbag	J4GI1.STCO2-.R--S-.-K--.---P-.P3-2	115	L	False	31/08/2015	
Anteriores Ibiza	Seat Ibiza	Respaldo-M. Almacenaje	J4GI1.PALK2-.R-QS-.-K--.--QP-.P3-3	125	R	False	31/08/2015	

Fig. 3.23 Ejemplo de formulario web de extracción de datos

A1 | Linea

	A	B	C	D	E	F	G	H	I
1	Linea	Modelo	Defecto	Broadcast Code	Secuencia	Mano	Fuera Secuencia	Fecha	Op
2	Anteriores Ibiza	Seat Ibiza	Respaldo-M. Almacenaje	J4BI1.MUTY0-.R--S-.----.-----.E3-3	109	R	False	31/08/2015	ELI
3	Anteriores Ibiza	Seat Ibiza	Respaldo-M. Almacenaje	J4BI1.KINK2-.R--S-.-K--.---P-.P3-2	110	R	False	31/08/2015	ELI
4	Anteriores Ibiza	Seat Ibiza	Respaldo-M. Almacenaje	J4BI1.MUTY0-.R--S-.----.-----.E3-3	113	R	False	31/08/2015	ELI
5	Anteriores Ibiza	Seat Ibiza	Ruteado-Ausencia brida negra cable airbag	J4GI1.STCO2-.R--S-.-K--.---P-.P3-2	115	L	False	31/08/2015	ELI
6	Anteriores Ibiza	Seat Ibiza	Respaldo-M. Almacenaje	J4GI1.PALK2-.R-QS-.-K--.--QP-.P3-3	125	R	False	31/08/2015	ELI
7	Anteriores Ibiza	Seat Ibiza	Respaldo-M. Almacenaje	J4GI1.STCO2-.R--S-.-K--.---P-.P3-2	131	R	False	31/08/2015	ELI
8	Anteriores Ibiza	Seat Ibiza	Respaldo-M. Almacenaje	J2BI1.MUTY0-.R--S-.----.-----.E3-3	136	R	False	31/08/2015	ELI
9	Anteriores Ibiza	Seat Ibiza	Cojín-M. Almacenaje	J4BI1.MUTY0-.R--S-.-K--.---P-.P3-3	143	L	False	31/08/2015	ELI
10	Anteriores Ibiza	Seat Ibiza	Respaldo-M. Almacenaje	J4BI1.MUTY0-.R--S-.-K--.---P-.P3-3	143	R	False	31/08/2015	ELI
11	Anteriores Ibiza	Seat Ibiza	Respaldo-M. Almacenaje	J4BI1.MUTY0-.R--S-.----.-----.E3-3	145	L	False	31/08/2015	ELI
12	Anteriores Ibiza	Seat Ibiza	Respaldo-M. Almacenaje	J2BI1.MUTY0-.R--S-.----.-----.E3-3	148	R	False	31/08/2015	ELI
13	Anteriores Ibiza	Seat Ibiza	Respaldo-M. Almacenaje	J4BI1.MUTY0-.R--S-.----.-----.E3-3	149	R	False	31/08/2015	ELI
14	Anteriores Ibiza	Seat Ibiza	Respaldo-M. Almacenaje	J4GI1.STCO2-.R--S-.-KC-.R--P-.P3-2	155	L	False	31/08/2015	ELI
15	Anteriores Ibiza	Seat Ibiza	Cojín-M. Almacenaje	J4BI1.MUTY0-.R--S-.-A--.---P-.E3-2	157	R	False	31/08/2015	ELI
16	Anteriores Ibiza	Seat Ibiza	Respaldo-M. Almacenaje	J2GI1.PALR2-.R--S-.-K--.RE-P-.P3-3	160	R	False	31/08/2015	ELI
17	Anteriores Ibiza	Seat Ibiza	Respaldo-M. Almacenaje	J2BD1.KINK2-.RE-P-.-K--.RE-S-.P3-3	166	R	False	31/08/2015	ELI
18	Anteriores Ibiza	Seat Ibiza	Respaldo-M. Almacenaje	J4GI1.FRPI2-.R--S-.-K--.R--P-.P3-2	173	R	False	31/08/2015	ELI
19	Anteriores Ibiza	Seat Ibiza	Trasrespaldo-M. Almacenaje	J4BD1.KINK2-.---P-.-K--.R--S-.P3-2	190	L	False	31/08/2015	ELI
20	Anteriores Ibiza	Seat Ibiza	Cojín-M. Almacenaje	J4BI1.KINK2-.R--S-.-KC-.R--P-.P3-2	191	L	False	31/08/2015	ELI

Fig. 3.24 Ejemplo de exportación de defectos a fichero Excel

En el desarrollo de nuestras pantallas de información en tiempo real se han creado 3 vistas que mostraban las producción del turno en curso, el top 5 de defectos de la hora en curso y el top 5 de defectos de la hora en curso respecto al total del turno. Estas tres vistas van rotando cada minuto de forma que se van mostrando por pantalla secuencialmente. Con el objetivo de no tener que esperar para ver el dato de una pantalla en concreto, se decide crear una cuarta vista que engloba todas las informaciones de forma resumida y que se actualiza cada minuto tal y como se puede observar en la figura 3.25. En el anexo referente a las modificaciones de desarrollo, podrán observarse las figuras de las 3 vistas sustituidas por la actual vista resumen.

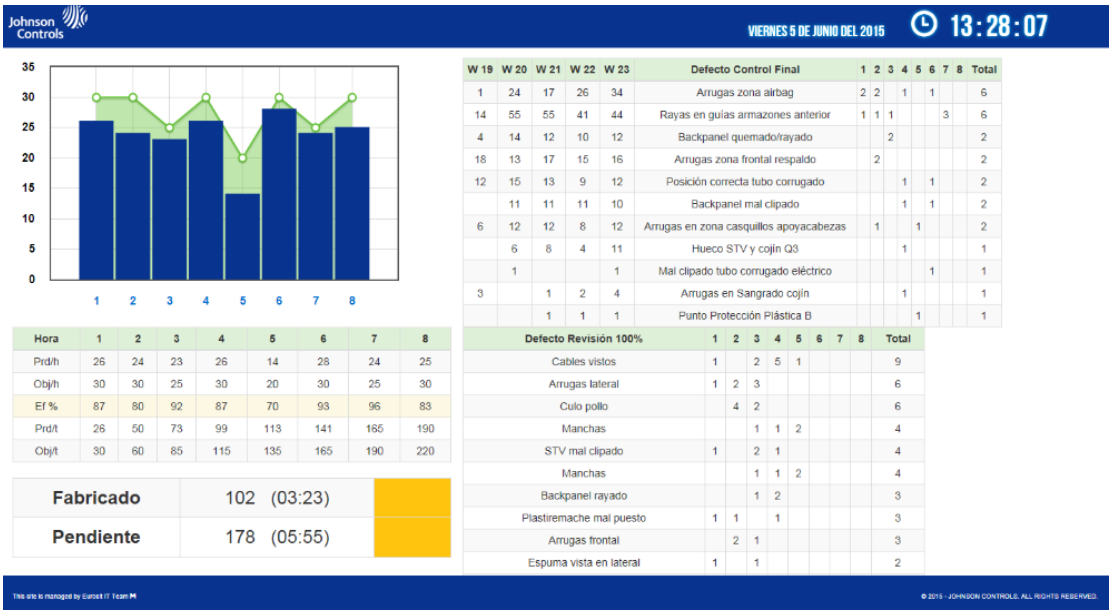


Fig. 3.25 Top 5 de defectos acumulados en la hora en curso

CAPÍTULO 4. CONCLUSIONES

Al ser un proyecto desarrollado dentro de una empresa, a lo largo de este capítulo final de conclusiones, se podrá ver el impacto real del trabajo desarrollado, cómo se han cumplido los objetivos marcados inicialmente, así como también otros que se han añadido tras su implementación y el impacto económico y medioambiental que ha supuesto.

Tras varios meses de la puesta en producción de la primera versión, y gracias a la colaboración de los departamentos y operarios que han trabajado con ella y sus modificaciones, se puede afirmar que se ha conseguido implementar una herramienta fiable, robusta y eficiente que cumple con todos los objetivos marcados en su fase de diseño.

A día de hoy, los departamentos de Calidad, Procesos y Producción disponen del registro de defectos de calidad en tiempo real y son capaces de actuar sobre los problemas que ocasionan estos defectos de una forma más rápida y eficiente. Según estimaciones del departamento de Calidad, se calcula que se han reducido entre un 11-14% los defectos de calidad detectados en el puesto de Inspección Visual. Esto supone una reducción considerable en el tiempo de retrabajo de estos asientos y una gran mejora en la calidad del producto acabado.

El desarrollo de esta herramienta no sólo ha cumplido con las expectativas de calidad acordadas con el cliente, sino que, mediante la implantación de la última aplicación diseñada, Quality Firewall, se ha buscado la excelencia en la calidad del producto acabado.

El proceso de trabajo se ha mejorado siguiendo los estándares de Johnson Controls, Inc. en cuanto a trabajo estandarizado, resolución de problemas, 5S, reducción de costes y control de errores.

Por otro lado, la puesta en producción de esta herramienta forma parte de una serie de medidas implantadas en las líneas de montaje con el objetivo de mejorar la eficiencia de su productividad. Esta herramienta ha contribuido en la reducción de los tiempos de procesos, la concienciación de los operarios en la calidad del producto y la mejora de los procesos de montajes. El departamento de procesos de J. C. Eurosit calcula un incremento en sus líneas de producción de un 85% a un 94% en el caso del Ibiza y de un 86% a un 94% en caso del Q3.

El desarrollo de esta herramienta ha supuesto un coste cercano a los 3.000$ y supone un ahorro anual de unos 25.000$. De estos 25.000$ de ahorro anuales se estiman unos 7.000$ relacionados con costes por garantías derivadas de quejas del cliente (ya sea SEAT o el comprador final) ante asientos en mal estado. En concepto de ahorro de costes asociados a la reducción de retrabajos por asientos con defectos detectados se calcula que la planta ahorra unos 8.000$

anuales. Finalmente, gracias a la implementación de esta herramienta se reduce el total de dinero perdido por la planta en material desechado en unos 11.000$ anuales.

Junto a la reducción mencionada anteriormente del material desechado por un montaje inadecuado, se ha de tener en cuenta el ahorro en papel e impresiones de los informes manuales previos a la digitalización de los defectos de calidad. Todo esto supone un impacto medioambiental mucho menor al ocasionado antes de la implantación de este sistema.

Para concluir con este capítulo, cabe destacar que las aplicaciones diseñadas en este TFC han sido presentadas a los departamentos centrales de Continuous Improvement de Johnson Controls, Inc. y ha sido considerada una Best Business Practice (BBP).

A su vez, J. C. Eurosit ha nominado a la herramienta para los Merit Award que la Unidad de Negocio Central otorga cuatrimestralmente a los procedimientos o las herramientas más innovadoras, que muestran un esfuerzo por encima de los deberes del trabajo normal, tienen un impacto medible en operaciones de la empresa y demuestran la excelencia en la conducción de la estrategia y los valores de Johnson Controls, Inc.

A nivel personal, estos premios y nominaciones hacen que me sienta muy satisfecho con el trabajo realizado y orgulloso de haber conseguido una herramienta eficaz y robusta, no solo para la planta de Abrera, sino para el resto de plantas JIT de Johnson Controls, Inc. a lo largo del mundo.

BIBLIOGRAFIA

[1] “MSDN API”

Disponible en: https://msdn.microsoft.com/library

[2] “W3Schools, HTML5 Tutorial”

Disponible en: http://www.w3schools.com/html/default.asp

[3] “W3C HTML5 API”

Disponible en: http://www.w3.org/TR/2014/REC-html5-20141028/

[4] “W3Schools, ASP.NET MVC Tutorial”

Disponible en: http://www.w3schools.com/aspnet/mvc_intro.asp

[5] “SQL Commands”

Disponible en: http://www.sqlcommands.net/

[6] “Coding Horror. A visual explanation of SQL Joins”

Disponible en: http://blog.codinghorror.com/a-visual-explanation-of-sql-joins/

[7] “Scribd. Estándares y consideraciones de desarrollo para Microsoft.NET”

Disponible en: http://es.scribd.com/doc/209660/Estandares-y-Consideraciones-de-Desarrollo-para-Microsoft-NET

[8] “Advosol, XML-DA.NET Server Toolkit”

Disponible en: https://advosol.com/pc-12-4-xml-da-net-server-toolkit.aspx

[9] “Siemens, Simatic S7-300”

Disponible en: http://w3.siemens.com/mcms/programmable-logic-controller/en/advanced-controller/s7-300/Pages/Default.aspx

[10] “Johnson Controls Eurosit, S. L.; Manual del Empleado, 2013”

ANEXOS

Anexo A. PLC Siemens

A diferencia de un ordenador normal, el PLC está diseñado para trabajar con múltiples señales de entrada y salida, rangos de temperatura ampliados, inmunidad al ruido eléctrico y resistencia a la vibración y al impacto.

Un PLC realiza, entre otras, las siguientes funciones:

- Recoger datos de las fuentes de entrada a través de las fuentes digitales y analógicas.
- Tomar decisiones en base a criterios preprogramados.
- Almacenar datos en la memoria.
- Generar ciclos de tiempo.
- Realizar cálculos matemáticos.
- Actuar sobre los dispositivos externos mediante las salidas analógicas y digitales.
- Comunicarse con otros sistemas externos.

En las figuras A.1 y A.2 se puede observar el modelo S7- 400 encargado de gestionar la línea de producción de los asientos del León. Este es el modelo más moderno instalado en J. C. Eurosit. El resto de líneas trabajan con un modelo S7-300, tal y como se puede observar en la figura A.3

Fig. A.1 PLC Siemens S7 – 400 Exterior

Fig. A.2 PLC Siemens S7 – 400

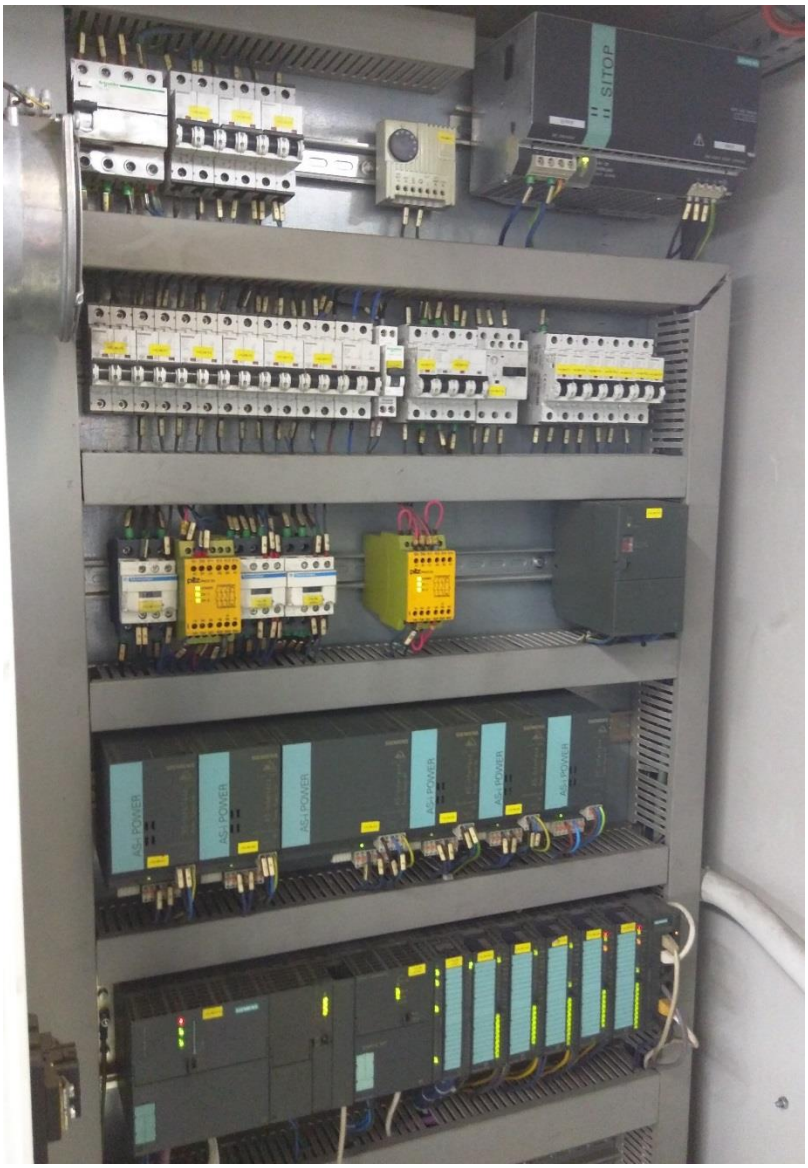


Fig. A.3 PLC Siemens S7 – 300

Tanto un modelo como otro se componen de una CPU, una fuente de alimentación, los módulos de entradas y salidas analógicas y digitales, módulos de funciones y el procesador de comunicaciones. Si bien es verdad que en la figura A.3 correspondiente al modelo 300 se ven más componentes encargados de otro tipo de funcionalidades de la línea. A su vez el modelo es menos compacto y más modular que su predecesor S7 – 400

Todo este hardware, una vez interconectado correctamente, se gestiona mediante aplicaciones software que acceden a los diferentes componentes a través de la conexión Ethernet del PLC. En la figura A.4 se muestra la aplicación NetPro, que se encarga de definir la configuración entre los diferentes módulos del PLC.

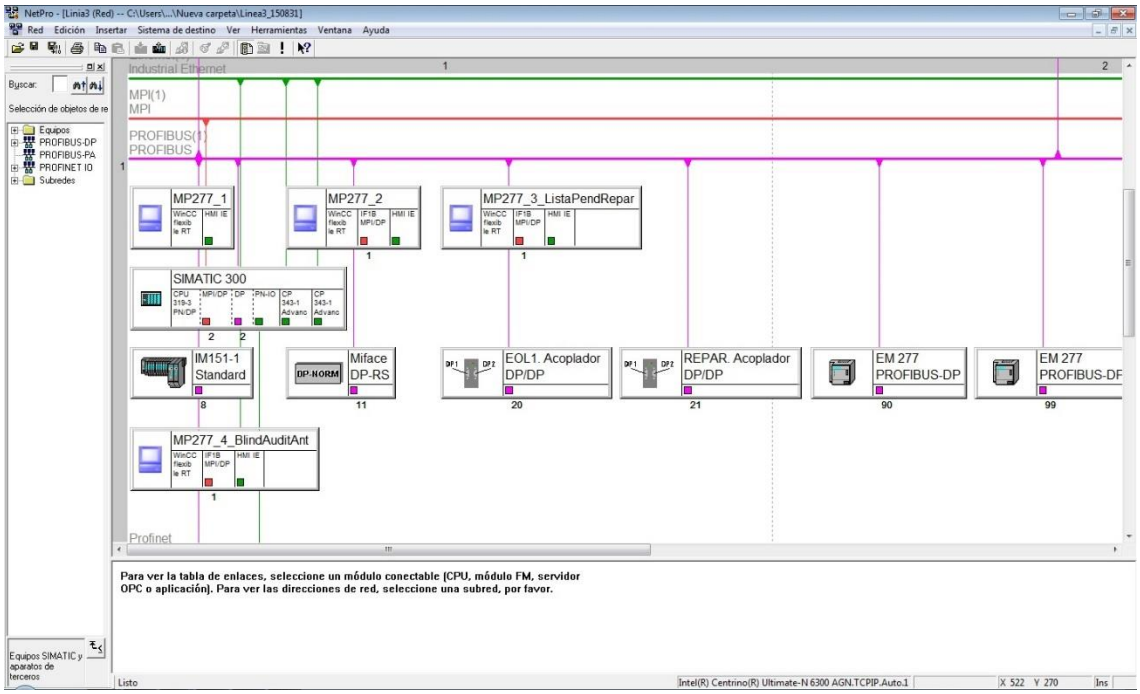


Fig. A.4 NetPro Proyecto Línea Q3 para PLC Siemens S7 – 300

Una vez establecida la configuración entre los diferentes módulos que integran el PLC se ha de proceder a la programación de las diferentes rutinas y funciones que el PLC ha de controlar a lo largo de la línea de producción.

La programación de un PLC Siemens S7 – 300 se compone de tres bloques básicos:

- Bloques de organización (OB)
- Bloques de función (FB)
- Bloques de memoria (DB)

Los bloques de organización constituyen el interface entre el sistema operativo y el programa de usuario. Son llamados por el sistema operativo y controlan el procesamiento cíclico y controlado por alarmas del programa, el comportamiento de arranque del sistema de automatización y el tratamiento de los errores. Programando los bloques de organización se define el comportamiento de la CPU.

Los bloques de función son los bloques programables. Contienen un programa que se ejecuta siempre que sea llamado. Básicamente son las funciones y rutinas programadas para cada tipo de funcionalidad u operación requerida a lo largo del proceso de montaje.

Por último, los bloques de memoria o DBs, son áreas de datos en las que se almacena la información de la línea de producción.

En la figura A.5 se muestra un ejemplo de la programación del PLC S7 – 300 de la línea de producción del Q3.

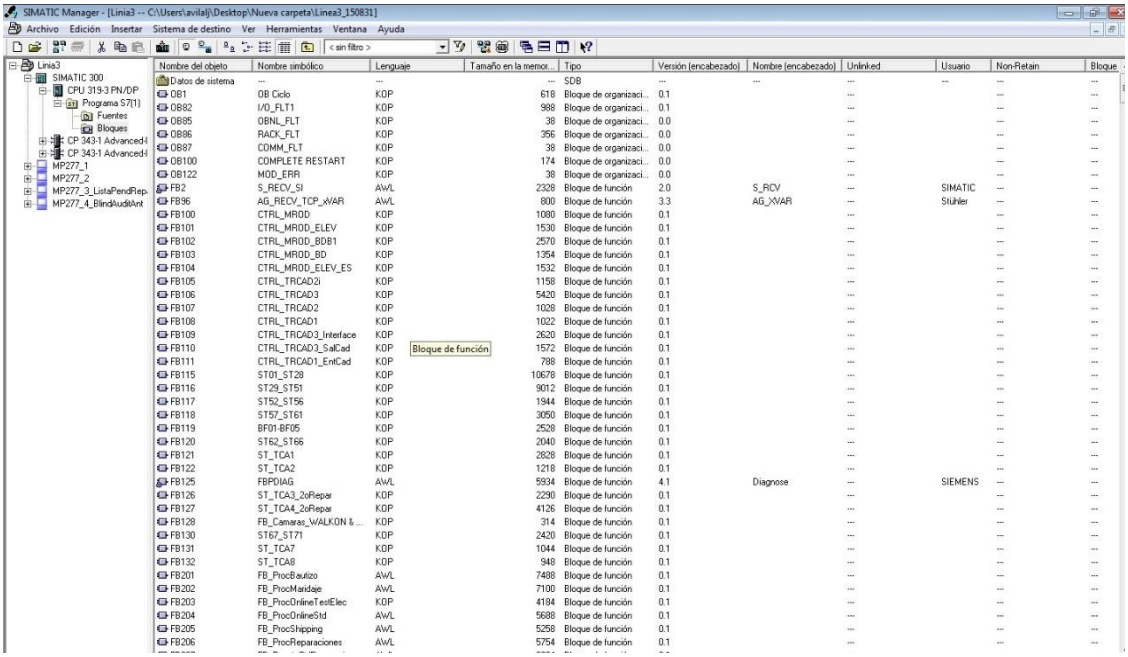


Fig. A.5 NetPro Proyecto Línea Q3 para PLC Siemens S7 – 300

Anexo B. Base de datos

En este anexo tiene como objetivo explicar, de forma sencilla y clara, el proceso de creación de bases de datos, tablas y procedimientos almacenados en un Servidor Microsoft SQL Server.

B.1. Creación de la base de datos

El proceso de creación de base de datos es tan sencillo como se muestra en las figuras que a aparecen a continuación.

En una primera instancia se ha de definir un nombre para la base de datos y asignarle un usuario propietario. Este usuario dispondrá, por su propietario, de todos los permisos para la gestión, administración y creación de elementos respecto a esa base de datos. Además de un nombre y un propietario, en SQL Server, se establece un tamaño inicial para la base de datos y su correspondiente fichero de log, así como el tipo de incremento una vez se alcance este tamaño inicial

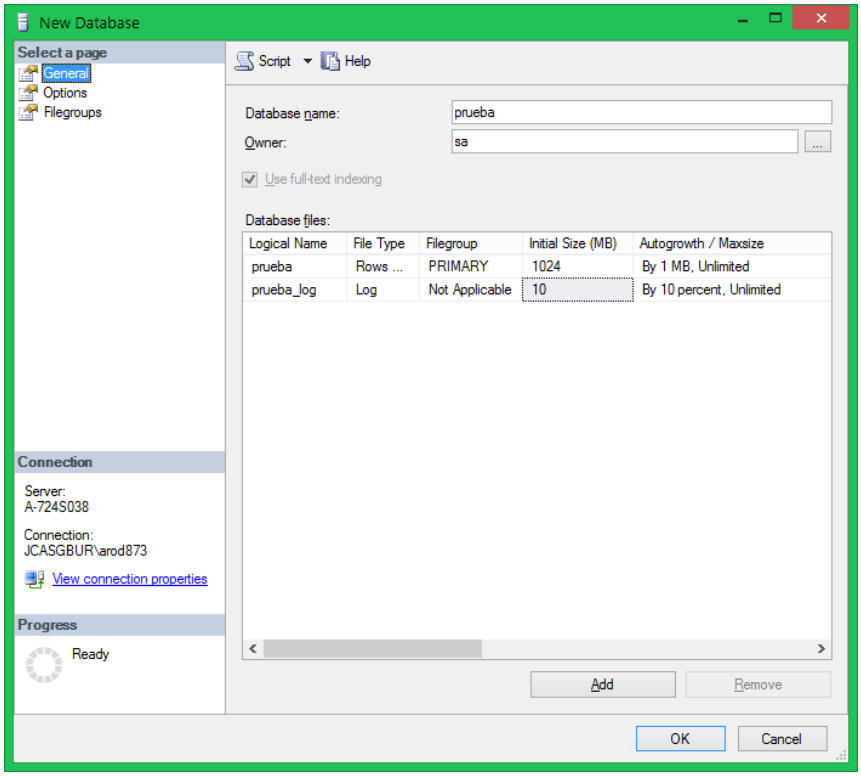


Fig. B.1 Creación de una nueva base de datos

Con estos datos ya se podría crear una base de datos de forma genérica. Si además queremos configurar otras opciones de forma detalla se puede acceder a las Opciones de creación, tal y como se muestra en la figura B.2.

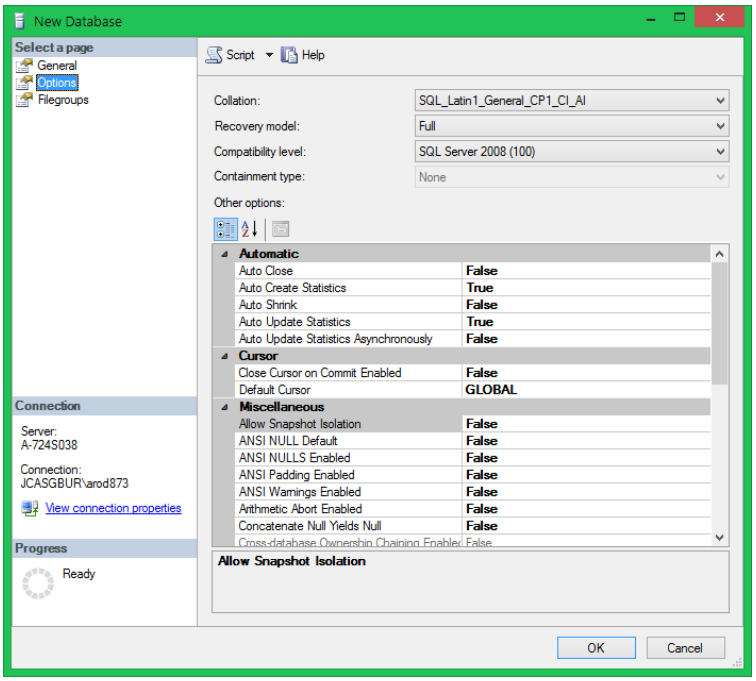


Fig. B.2 Creación de una nueva base de datos

Adicionalmente a estas configuraciones y por motivos de seguridad, se asociará un usuario a la nueva base de datos restringiéndole los permisos. Este usuario será el que se usará para establecer comunicación entre las aplicaciones de escritorio y las páginas web con la base de datos. En nuestro caso se le darán únicamente permisos para conectar a la base de datos, ejecutar consultas de selección, creación y actualización y la ejecución de procedimientos almacenados. De esta forma se minimiza el riesgo de que, cualquier fallo o error de las aplicaciones que conectan a la base de datos, puedan borrar datos, tablas o causar cualquier otra modificación para la que no estén diseñadas.

En la figura B.3 se muestra un ejemplo de este tipo de configuración.

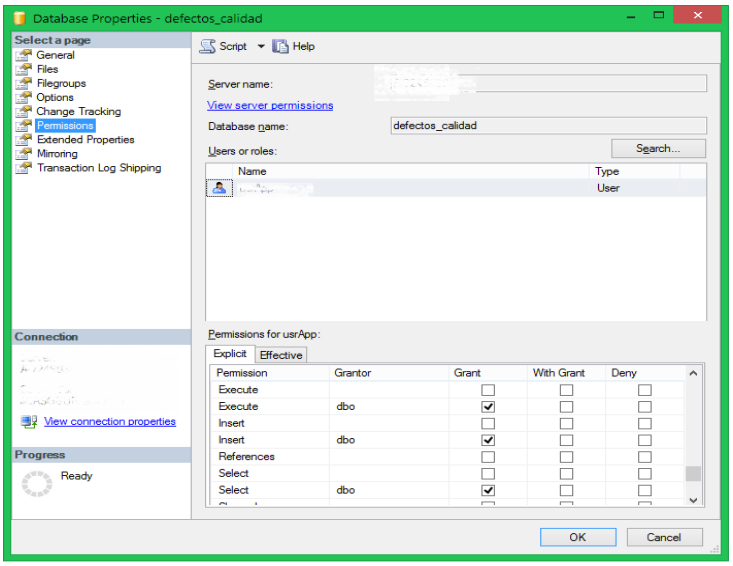


Fig. B.3 Asignación de permisos por usuario a base de datos

B.2. Tablas

Una vez creada la base de datos se puede empezar a crear las tablas en las que se almacenarán los datos necesarios para el correcto funcionamiento de las aplicaciones de escritorio.

El proceso de creación de tablas en SQL Server mediante su interfaz gráfica es sencillo y rápido. Como se puede observar en la figura B.4, únicamente se ha de definir el nombre con el que se identificará a la tabla en cuestión y las columnas en las que se almacenarán los datos.

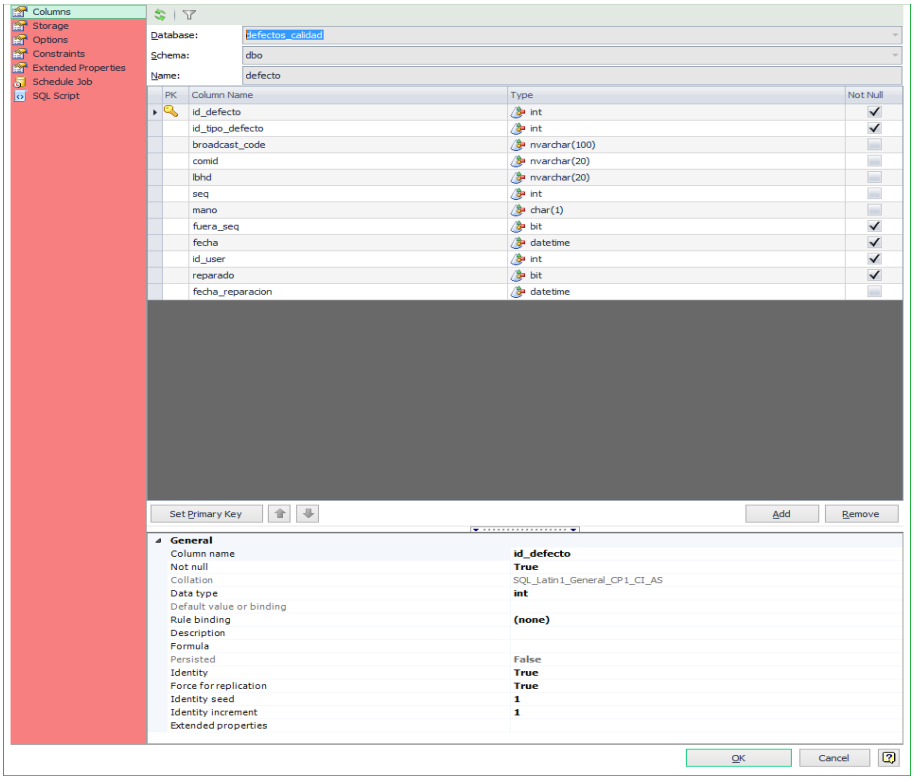
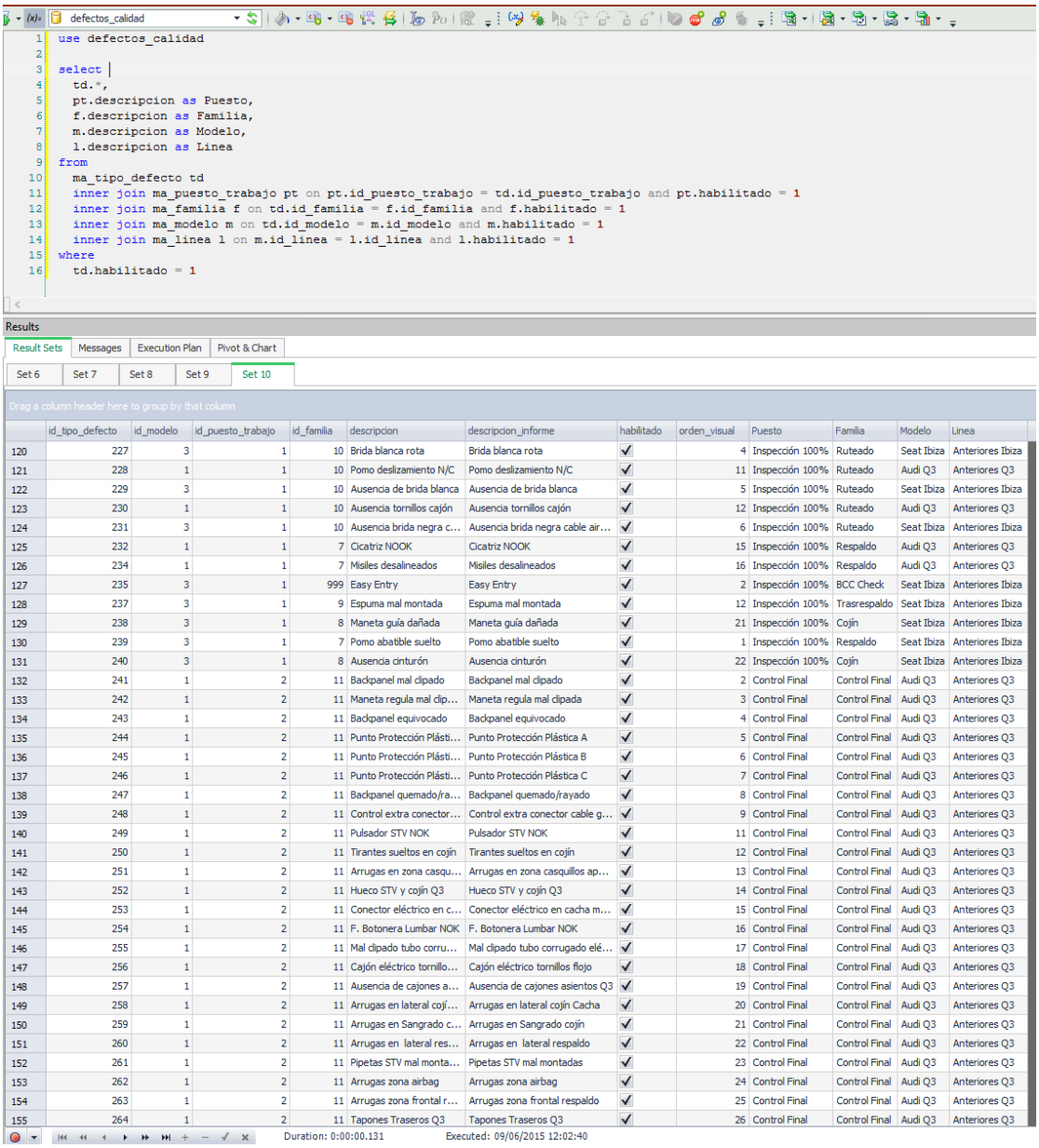


Fig. B.4 Creación de una tabla SQL

Cada columna tendrá un tipo de dato asociado así como otro tipo de especificaciones más detalladas. Se puede establecer si la columna en cuestión será una clave primaria, permitirá o no valores nulos e incluso definir un tipo de valor por defecto o un rango o mascara de valores aceptados. Según las necesidades de diseño y el grado de detalle y control que se quiera definir sobre la base de datos se podrá, o no, configurar todo este tipo de opciones.

En la figura B.5 se muestra un ejemplo de consulta en la que se relacionan las diferentes tablas implementadas de forma que se obtiene un listado de defectos por línea de montaje, puesto de trabajo y modelo de coche.



```
use defectos_calidad

select
  td.*,
  pt.descripcion as Puesto,
  f.descripcion as Familia,
  m.descripcion as Modelo,
  l.descripcion as Linea
from
  ma_tipo_defecto td
  inner join ma_puesto_trabajo pt on pt.id_puesto_trabajo = td.id_puesto_trabajo and pt.habilitado = 1
  inner join ma_familia f on td.id_familia = f.id_familia and f.habilitado = 1
  inner join ma_modelo m on td.id_modelo = m.id_modelo and m.habilitado = 1
  inner join ma_linea l on m.id_linea = l.id_linea and l.habilitado = 1
where
  td.habilitado = 1
```

Results

Result Sets | Messages | Execution Plan | Pivot & Chart

Set 6 | Set 7 | Set 8 | Set 9 | Set 10

Drag a column header here to group by that column

	id_tipo_defecto	id_modelo	id_puesto_trabajo	id_familia	descripcion	descripcion_informe	habilitado	orden_visual	Puesto	Familia	Modelo	Linea
120	227	3	1	10	Brida blanca rota	Brida blanca rota	✓	4	Inspección 100%	Ruteado	Seat Ibiza	Anteriores Ibiza
121	228	1	1	10	Pomo deslizamiento N/C	Pomo deslizamiento N/C	✓	11	Inspección 100%	Ruteado	Audi Q3	Anteriores Q3
122	229	3	1	10	Ausencia de brida blanca	Ausencia de brida blanca	✓	5	Inspección 100%	Ruteado	Seat Ibiza	Anteriores Ibiza
123	230	1	1	10	Ausencia tornillos cajón	Ausencia tornillos cajón	✓	12	Inspección 100%	Ruteado	Audi Q3	Anteriores Q3
124	231	3	1	10	Ausencia brida negra c...	Ausencia brida negra cable air...	✓	6	Inspección 100%	Ruteado	Seat Ibiza	Anteriores Ibiza
125	232	1	1	7	Cicatriz NOOK	Cicatriz NOOK	✓	15	Inspección 100%	Respaldo	Audi Q3	Anteriores Q3
126	234	1	1	7	Misiles desalineados	Misiles desalineados	✓	16	Inspección 100%	Respaldo	Audi Q3	Anteriores Q3
127	235	3	1	999	Easy Entry	Easy Entry	✓	2	Inspección 100%	BCC Check	Seat Ibiza	Anteriores Ibiza
128	237	3	1	9	Espuma mal montada	Espuma mal montada	✓	12	Inspección 100%	Trasrespaldo	Seat Ibiza	Anteriores Ibiza
129	238	3	1	8	Maneta guía dañada	Maneta guía dañada	✓	21	Inspección 100%	Cojín	Seat Ibiza	Anteriores Ibiza
130	239	3	1	7	Pomo abatible suelto	Pomo abatible suelto	✓	1	Inspección 100%	Respaldo	Seat Ibiza	Anteriores Ibiza
131	240	3	1	8	Ausencia cinturón	Ausencia cinturón	✓	22	Inspección 100%	Cojín	Seat Ibiza	Anteriores Ibiza
132	241	1	2	11	Backpanel mal clipado	Backpanel mal clipado	✓	2	Control Final	Control Final	Audi Q3	Anteriores Q3
133	242	1	2	11	Maneta regula mal clip...	Maneta regula mal clipada	✓	3	Control Final	Control Final	Audi Q3	Anteriores Q3
134	243	1	2	11	Backpanel equivocado	Backpanel equivocado	✓	4	Control Final	Control Final	Audi Q3	Anteriores Q3
135	244	1	2	11	Punto Protección Plásti...	Punto Protección Plástica A	✓	5	Control Final	Control Final	Audi Q3	Anteriores Q3
136	245	1	2	11	Punto Protección Plásti...	Punto Protección Plástica B	✓	6	Control Final	Control Final	Audi Q3	Anteriores Q3
137	246	1	2	11	Punto Protección Plásti...	Punto Protección Plástica C	✓	7	Control Final	Control Final	Audi Q3	Anteriores Q3
138	247	1	2	11	Backpanel quemado/ra...	Backpanel quemado/rayado	✓	8	Control Final	Control Final	Audi Q3	Anteriores Q3
139	248	1	2	11	Control extra conector...	Control extra conector cable g...	✓	9	Control Final	Control Final	Audi Q3	Anteriores Q3
140	249	1	2	11	Pulsador STV NOK	Pulsador STV NOK	✓	11	Control Final	Control Final	Audi Q3	Anteriores Q3
141	250	1	2	11	Tirantes sueltos en cojín	Tirantes sueltos en cojín	✓	12	Control Final	Control Final	Audi Q3	Anteriores Q3
142	251	1	2	11	Arrugas en zona casqu...	Arrugas en zona casquillos ap...	✓	13	Control Final	Control Final	Audi Q3	Anteriores Q3
143	252	1	2	11	Hueco STV y cojín Q3	Hueco STV y cojín Q3	✓	14	Control Final	Control Final	Audi Q3	Anteriores Q3
144	253	1	2	11	Conector eléctrico en c...	Conector eléctrico en cacha m...	✓	15	Control Final	Control Final	Audi Q3	Anteriores Q3
145	254	1	2	11	F. Botonera Lumbar NOK	F. Botonera Lumbar NOK	✓	16	Control Final	Control Final	Audi Q3	Anteriores Q3
146	255	1	2	11	Mal clipado tubo corru...	Mal clipado tubo corrugado elé...	✓	17	Control Final	Control Final	Audi Q3	Anteriores Q3
147	256	1	2	11	Cajón eléctrico tornillo...	Cajón eléctrico tornillos flojo	✓	18	Control Final	Control Final	Audi Q3	Anteriores Q3
148	257	1	2	11	Ausencia de cajones a...	Ausencia de cajones asientos Q3	✓	19	Control Final	Control Final	Audi Q3	Anteriores Q3
149	258	1	2	11	Arrugas en lateral cojí...	Arrugas en lateral cojín Cacha	✓	20	Control Final	Control Final	Audi Q3	Anteriores Q3
150	259	1	2	11	Arrugas en Sangrado c...	Arrugas en Sangrado cojín	✓	21	Control Final	Control Final	Audi Q3	Anteriores Q3
151	260	1	2	11	Arrugas en lateral res...	Arrugas en lateral respaldo	✓	22	Control Final	Control Final	Audi Q3	Anteriores Q3
152	261	1	2	11	Pipetas STV mal monta...	Pipetas STV mal montadas	✓	23	Control Final	Control Final	Audi Q3	Anteriores Q3
153	262	1	2	11	Arrugas zona airbag	Arrugas zona airbag	✓	24	Control Final	Control Final	Audi Q3	Anteriores Q3
154	263	1	2	11	Arrugas zona frontal r...	Arrugas zona frontal respaldo	✓	25	Control Final	Control Final	Audi Q3	Anteriores Q3
155	264	1	2	11	Tapones Traseros Q3	Tapones Traseros Q3	✓	26	Control Final	Control Final	Audi Q3	Anteriores Q3

Duration: 0:00:00.131 Executed: 09/06/2015 12:02:40

Fig. B.5 Ejemplo de una consulta de tablas relacionadas

B.3. Procedimientos almacenados

Un procedimiento almacenado no es más que un grupo de instrucciones SQL que permite aceptar parámetros de entrada y devolver información, lo que hace posible añadir subrutinas propias y procedimientos para las bases de datos. Se utilizarán este tipo de procedimientos para no definir dentro del código fuente de las aplicaciones de escritorio las instrucciones de manejo de datos SQL.

En la figura B.6 se muestra un ejemplo de creación de procedimiento almacenado. En el tan sólo se ha de definir un nombre, establecer los parámetros de entrada y/o salida y definir la rutina T-SQL a realizar.

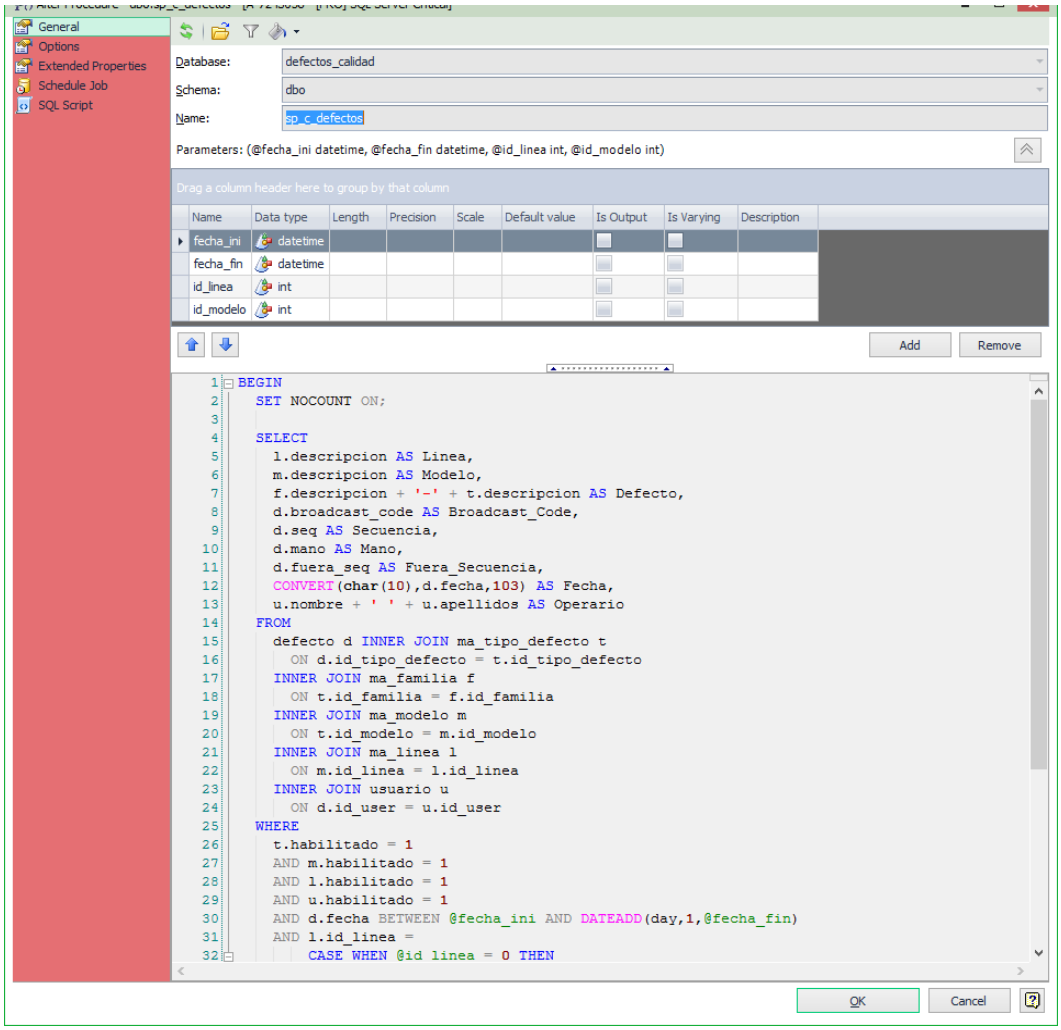


Fig. B.6. Ejemplo de creación de procedimiento almacenado

Anexo C. Creación de etiquetas de lectura OPC

Como ya se ha introducido en el capítulo 3.2, la estructura de datos de un servidor OPC se compone de canales, componentes, grupos de etiquetas y etiquetas.

Mediante un canal se establece una canal de comunicación a un tipo específico de dispositivo OPC, ya sea una atornilladora, un PLC o cualquier otro tipo de componente que permita este tipo de protocolo. En el momento de crear el canal se define el tipo de dispositivo que se asociará a dicho canal y los parámetros de configuración que requiera. En este caso en concreto se usará una conexión Siemens TCP/IP Ethernet con permisos de lectura y escritura.

Asociado a este canal se crean los dispositivos. Un dispositivo es la definición de un tipo de conexión directa con un dispositivo OPC en concreto. Tal y como se observan en las capturas de pantalla mostradas a continuación, en el momento de la creación se le define un nombre, un tipo de dispositivo relacionado con el canal sobre el que lo estamos creando, una dirección IP y una serie de configuraciones respecto al tiempo de respuesta, retardos, puertos de escucha y demás especificaciones técnicas.

Mediante las figuras C.1, C.2, C.3, C.4 y C.5 se puede observar el proceso de creación de un dispositivo OPC paso a paso

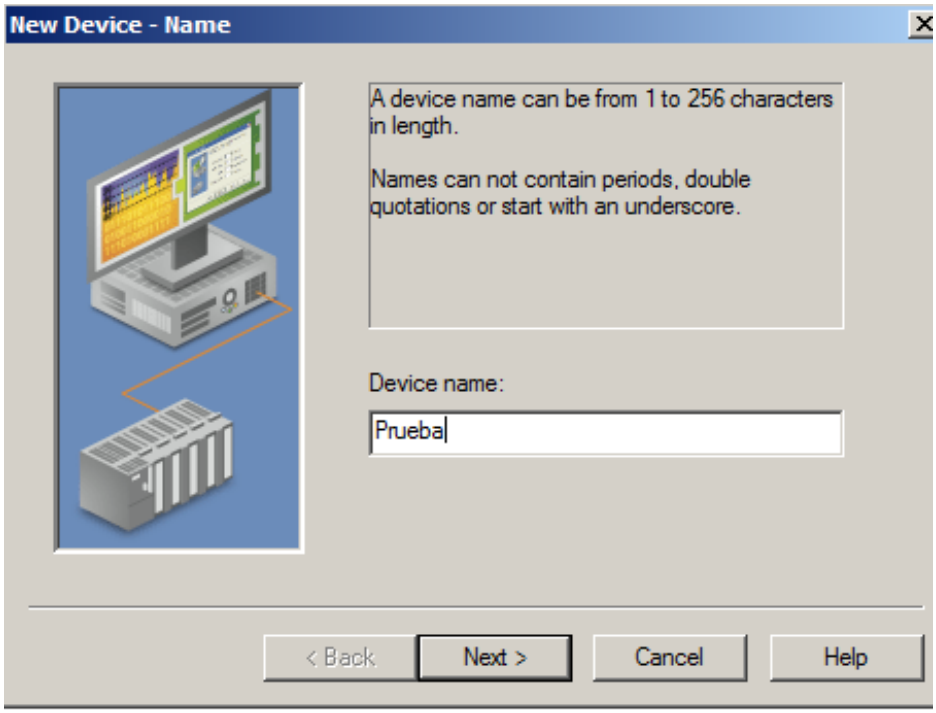


Fig. C.1 Creación de un dispositivo OPC, paso 1

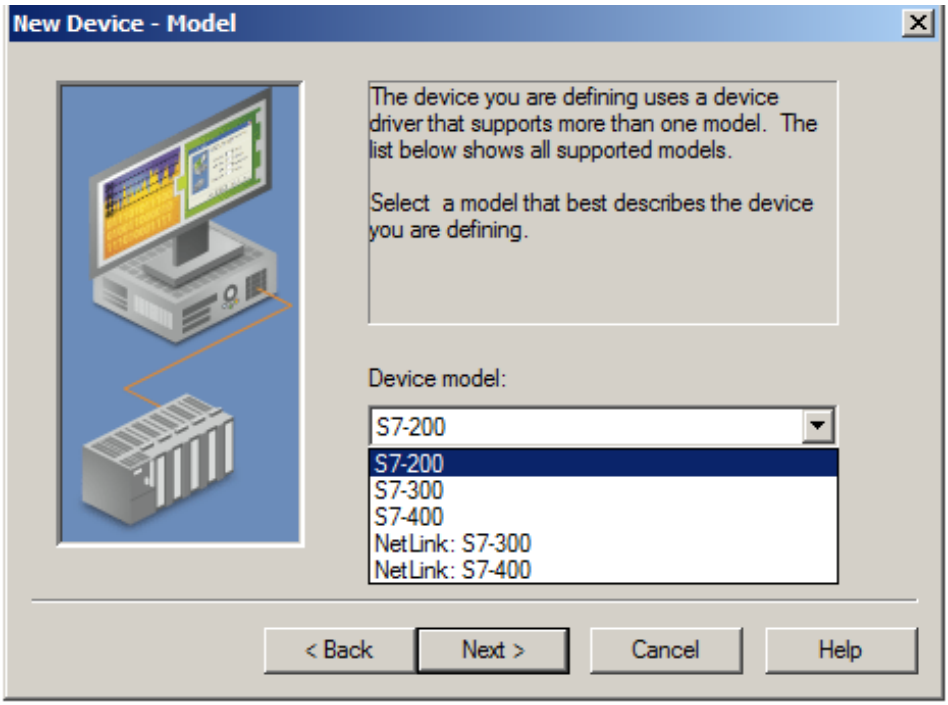


Fig. C.2 Creación de un dispositivo OPC, paso 2

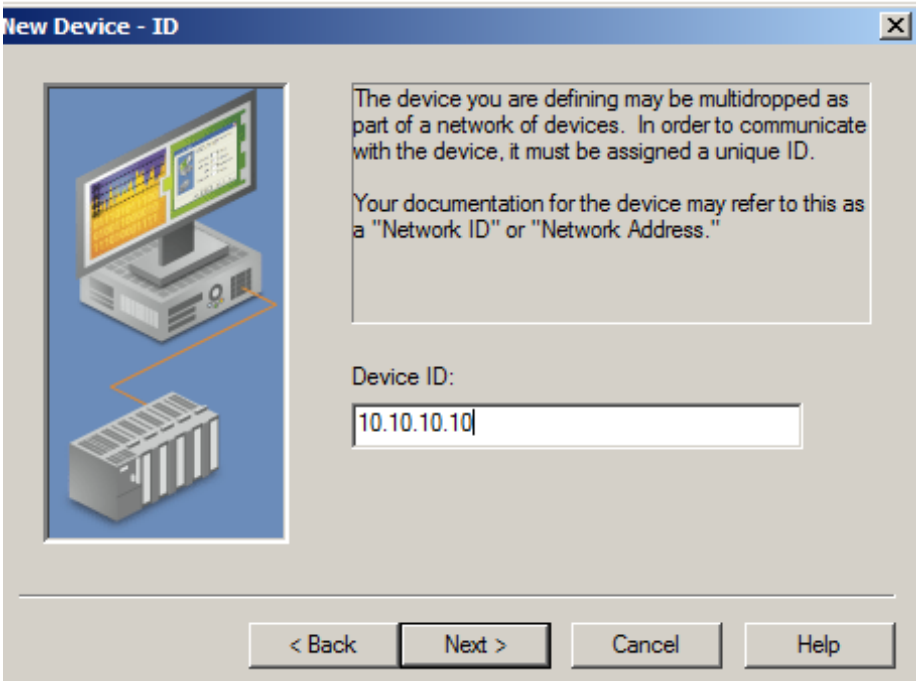


Fig. C.3 Creación de un dispositivo OPC, paso 3

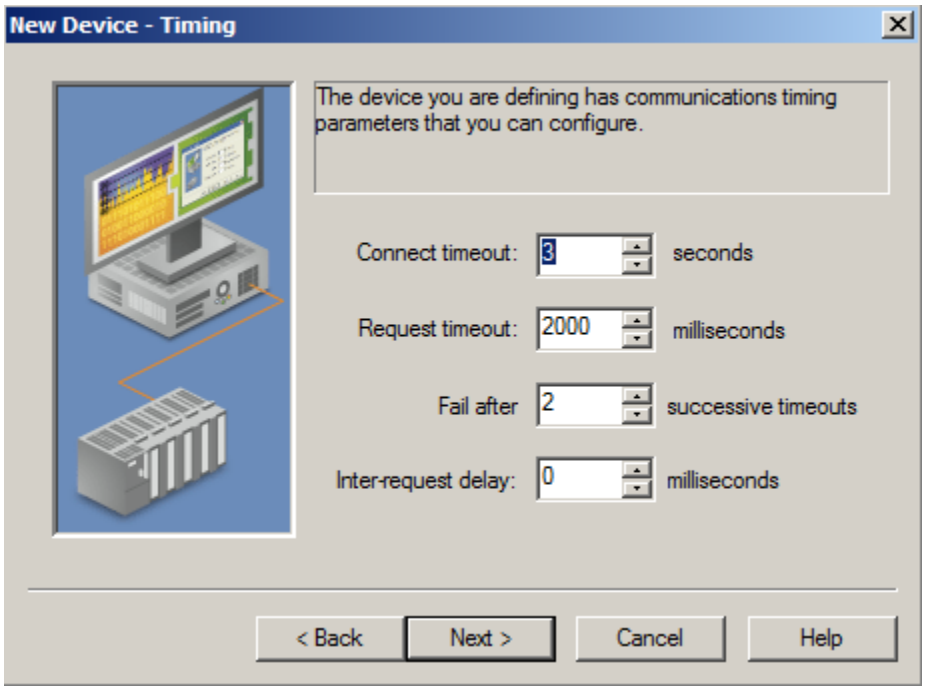


Fig. C.4 Creación de un dispositivo OPC, paso 4

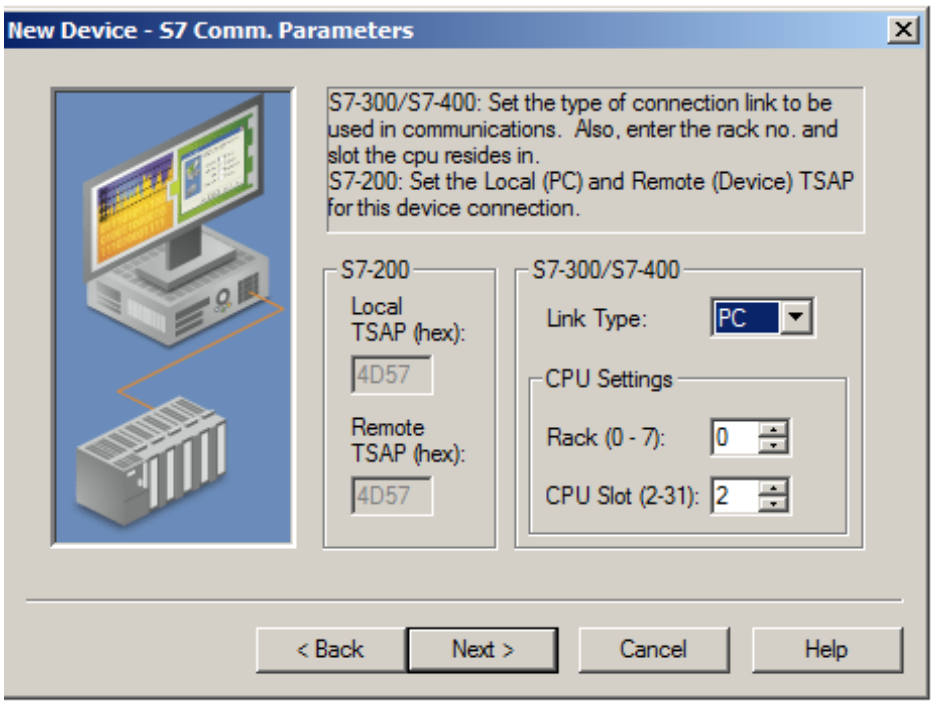


Fig. C.5 Creación de un dispositivo OPC, paso 5

Sobre cada dispositivo definido se pueden crear directamente las etiquetas o bien agruparlos en grupos de etiquetas (que no son más que carpetas en las que asociar grupos).

En la creación de la etiqueta únicamente se indicará el nombre con él se le identificará, la dirección del DB al que se le asociará, el tipo de dato que almacena y si será una etiqueta de lectura, escritura o ambas cosas. En la figura C.6 se puede observar un ejemplo de etiqueta.

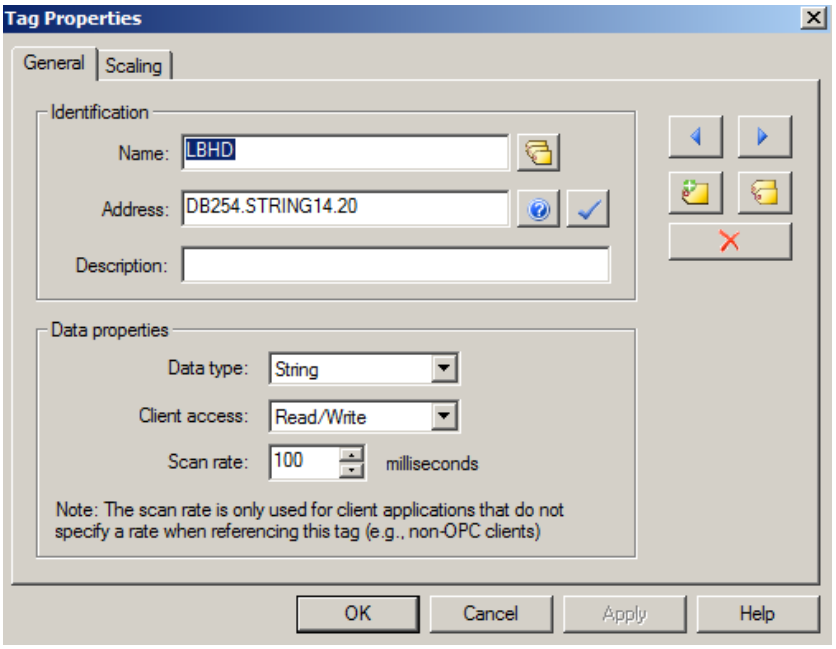


Fig. C.6 Creación de una etiqueta OPC

Según el tipo de dato al que se quiera acceder la dirección de la DB se escribirá de una forma u otra. Si es una estructura definida como un carácter (8 bits), un número de 32 bits (entero) o un número de 64 bits (doble) únicamente indicaremos su posición dentro del DB. Si por el contrario es una estructura variable como puede ser una cadena de caracteres, indicaremos su posición inicial dentro del DB seguido de un punto y su longitud en bytes. A continuación se muestra un ejemplo de sintaxis extraído del manual de OPC Server

DB<num>.<TipoDato><dirección>
DB<num>.<TipoDato><dirección>.<bit>
DB<num>.<TipoDato><dirección>.<longitud>

Los tipos de datos utilizados en este proyecto son ‘STRING’, ‘C’, ‘I’ y ‘D’, sin embargo hay muchas más definiciones de datos posibles en el estándar OPC.

Anexo D. Fichero de configuración

Como ya se ha comentado en el apartado 3.3.1, en este anexo se puede observar un ejemplo de fichero de configuración para las aplicaciones de escritorio desarrolladas en este TFC.

```
<?xml version="1.0"?>
<configuration>
  <startup>
    <supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.5.2"/>
  </startup>
  <appSettings>
    <clear/>
    <add key="Entorno" value="1"/>
    <add key="InputMode" value="True"/>
    <add key="LineaID" value="1"/>
    <add key="GodModePass" value="[redacted]"/>
    <add key="OPCStationL1" value="Channel14.A.A_ONL_CAL.CtrlProc"/>
    <add key="OPCStationL2" value="Channel15.B.B_ONL_CAL.CtrlProc"/>
    <add key="OPCStationL3" value="Channel1.G.G_ONL_CAL.CtrlProc"/>
    <add key="ClientSettingsProvider.ServiceUri" value=""/>
  </appSettings>
  <connectionStrings>
    <clear/>
    <add name="DBConn_Pro" connectionString="d2NZT+Bv5qKhMPpDDpz5ckQRG[redacted]
    <add name="DBConn_Des" connectionString="d2NZT+Bv5qKhMPpDDpz5cgyDz[redacted]
    <add name="OPCServer_Pro" connectionString="HVA4ywGuan+tXpdGN8B3Tr[redacted]
    <add name="OPCServer_Des" connectionString="HVA4ywGuan+tXpdGN8B3Tr[redacted]
  </connectionStrings>
  <system.web>
    <membership defaultProvider="ClientAuthenticationMembershipProvider">
      <providers>
        <add name="ClientAuthenticationMembershipProvider"
        type="System.Web.ClientServices.Providers.ClientFormsAuthenticationMembershipProvider,
        System.Web.Extensions, Version=4.0.0.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35"
        serviceUri=""/>
      </providers>
    </membership>
    <roleManager defaultProvider="ClientRoleProvider" enabled="true">
      <providers>
        <add name="ClientRoleProvider" type="System.Web.ClientServices.Providers.ClientRoleProvider,
        System.Web.Extensions, Version=4.0.0.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35"
        serviceUri="" cacheTimeout="86400"/>
      </providers>
    </roleManager>
  </system.web>
</configuration>
```

Fig. D.1 Ejemplo de fichero de configuración

Lo primero que se puede observar en el fichero de configuración en el tag <supportedRuntime> es la versión del Framework .NET necesario para ejecutar la aplicación.

A continuación, en el tag <appSettings> se definen las diferentes variables de entorno de la aplicación:

- Entorno: 0 – Desarrollo, 1 – Producción
- InputMode: 0 – Modo Escaner, 1 – Modo OPC
- LineaID : 1 – Ibiza, 2 – León, 3 – Q3
- GodModePass: Contraseñar cifrada mediante MD5 Hash para acceder al modo administrador de la aplicación
- OPCStationLx: Cadena de conexión correspondiente a los dispositivos vinculados a los PLCs de cada línea.

El otro tag remarcable corresponde a <connectionStrings> el cual contiene las cadenas de conexión, cifradas mediante MD5 Hash, hacia las bases de datos y los servidores OPC de desarrollo y producción.

Anexo E. ASP Web Forms vs ASP.NET MVC

En el apartado 3.4 se habla sobre la coexistencia de dos entornos web dentro de la misma intranet de J. C. Eurosit. En este anexo se explicarán las diferencias entre los dos entornos de programación utilizados en el desarrollo de este TFC.

La principal diferencia entre ASP Web Forms y ASP.Net MVC es el concepto de programación que cada tipo de tecnología define.

ASP Web Forms orienta su programación en un modelo formulario, muy similar al modelo de programación clásico de Visual Basic. Se basa en una programación manejada por eventos y de desarrollo rápida gracias a la utilización de los controles de servidor. Estos controles son muy similares a los de las aplicaciones de escritorio utilizados en VB.Net y, de igual forma, nos permiten asociar bloques de códigos a la serie de eventos que son capaces de manejar. Otra de las grandes ventajas de este tipo de programación es la pequeña curva de aprendizaje para aquellos programadores que tengan pocos conocimientos HTML y JavaScript. Al ser un entorno prácticamente idéntico a las aplicaciones VB.Net la metodología de programación es muy similar.

Una página web se compone, tal y como se puede ver en la figura E.1, de su código de diseño *.aspx y su denominado Code Behind, figura E.2, que es el archivo que contiene el código que se ejecutará en el servidor web. Su extensión será *.aspx.vb o *.aspx.cs, dependiendo de si se trabaja con VB.NET o C# en la sintaxis de Code Behind.

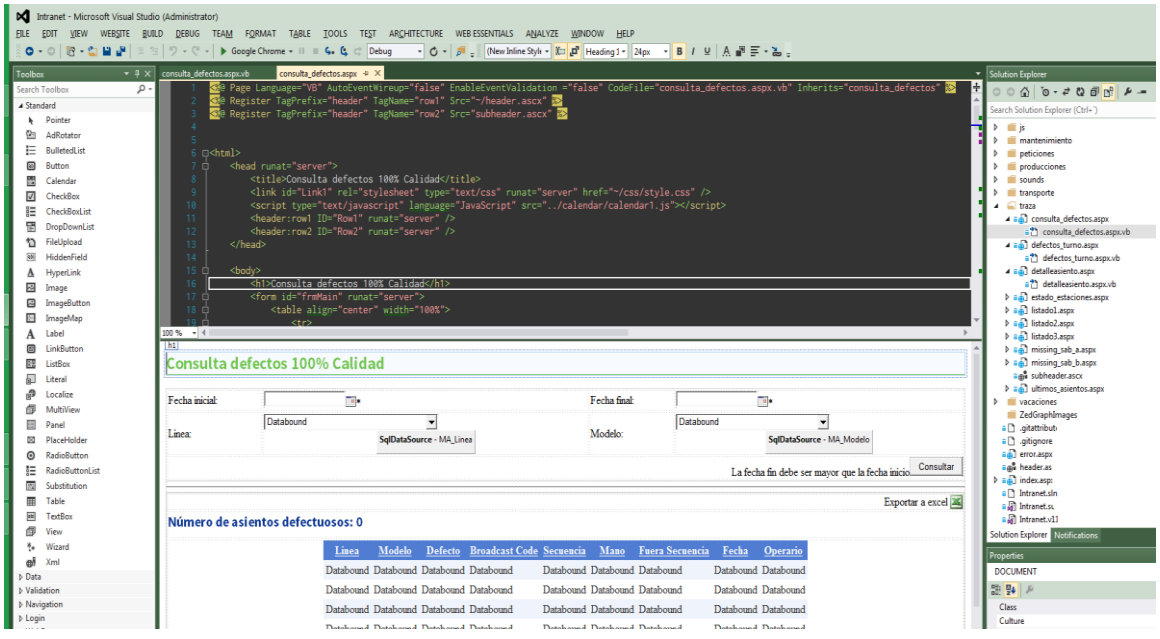


Fig. E.1 Ejemplo de programación en ASP Web Forms. Código de diseño

Como se puede observar en la figura E.1 se dispone de un código de diseño formado por un formulario web que contiene controles muy similares a los utilizados en la programación de aplicaciones de escritorio en Visual Basic. En

la parte superior se puede ver la traducción automática que hace el IDE de programación de este formulario de diseño a lengua HTML que será el que interprete el servidor web.

En la primera línea del código se puede ver la instrucción que asocia, este formulario de diseño con su Code Behind.

```
<%@ Page Language="VB" AutoEventWireup="false"
EnableEventValidation ="false"
CodeFile="consulta_defectos.aspx.vb"
Inherits="consulta_defectos" %>
```

Esta instrucción establece que todas las funciones programadas en el fichero `consulta_defectos.aspx.vb` están asociadas a ese formulario. En la figura E.2 podemos observar un ejemplo de código correspondiente a ese fichero Code Behind

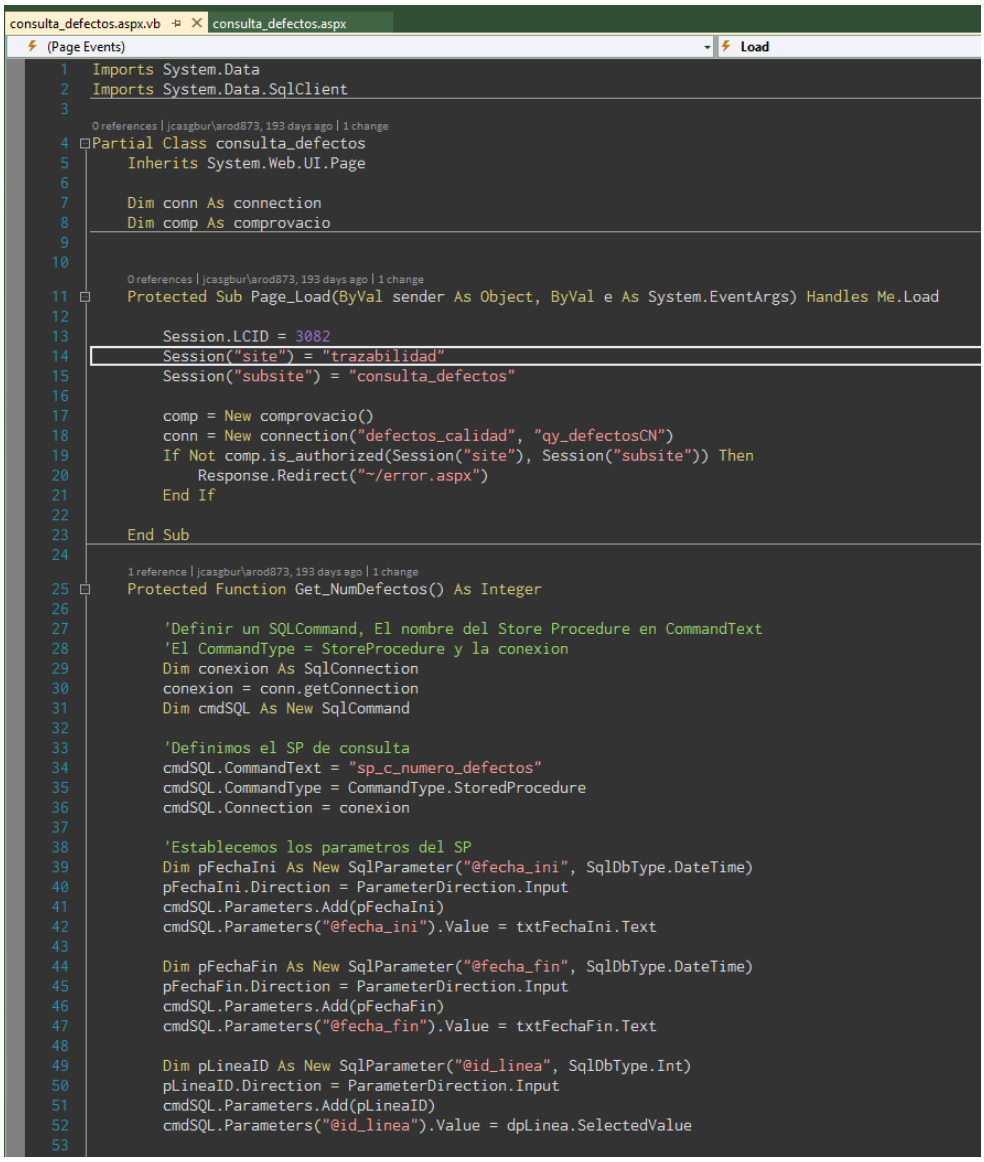


Fig. E.2 Ejemplo de programación en ASP Web Forms. Code Behind

En este caso se puede ver la programación asociada al evento de la carga de la página, en la cual se establecen una serie de valores de iniciación así como se define una conexión a la base de datos para establecer las consultas correspondientes. En la otra parte del código se puede ver parte de la función encargada de llamar al procedimiento almacenado que devuelve el número de defectos registrados para un determinado intervalo de fechas.

Si bien la programación en ASP Web Forms es rápida y sencilla, cuenta con la principal desventaja de tener poco control sobre el código HTML generado. El código incluido en el Code Behind no es fácilmente reutilizable en otro Web Form y no genera URLs amigables, lo cual afecta al SEO (posicionamiento en buscadores) de la página.

Por otro lado ASP.Net MVC está diseñado para trabajar según el patrón Modelo Vista Controlador. Este tipo de programación separa los datos y la lógica de negocio de una aplicación del interfaz de usuario. La programación basada en MVC se compone de tres tipos de elementos:

- Modelo: Es la capa donde se trabaja con los datos. Los datos los se encontrarán habitualmente en una base de datos, por lo que en los modelos se definen todas las funciones que accederán a las tablas y harán las consultas necesarias para acceder a ellos y manipularlos.
- Vista: Es la representación del modelo de datos. Como su nombre nos da a entender, contienen el código de nuestra aplicación que va a producir la visualización de las interfaces de usuario. Las vistas requieren los datos de los modelos y ellas generan la salida, sin embargo no realizan un acceso directo a estos.
- Controlador: El controlador contiene el código necesario para responder a las acciones que se solicitan en la aplicación. Es una capa que sirve de enlace entre los modelos y las vistas.

En la figura E.3 se puede observar la relación entre los tres elementos de la programación MVC

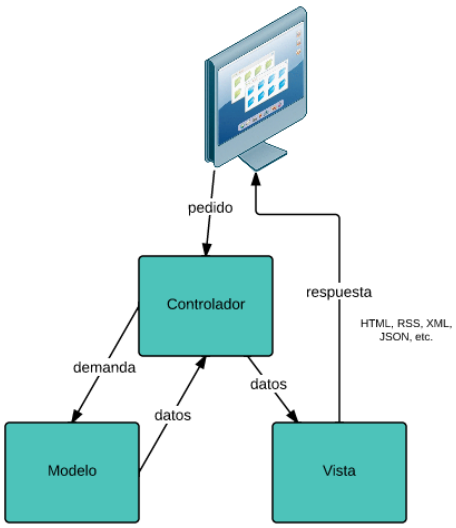


Fig. E.3 Patrón Modelo Vista Controlador

Las principales ventajas de una programación según el patrón MVC son una arquitectura de proyecto ordenada, una gran capacidad para la reutilización de código y un control total sobre el código HTML. La desventaja es que la curva de aprendizaje es mucho más lenta y costosa.

Como ya se ha comentado anteriormente, las pantallas de información en tiempo real ya han sido migradas a ASP.Net MVC. En la figura E.4 se puede observar la estructura de proyecto de la nueva intranet en la cual se integran estas pantallas de información.

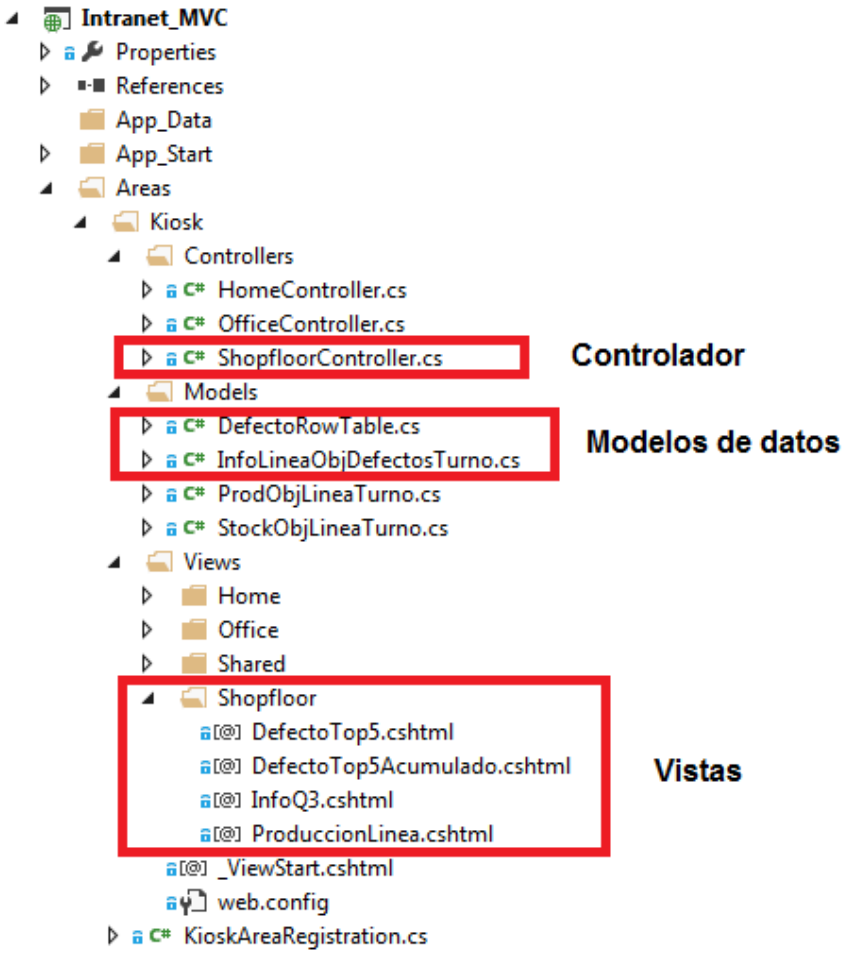


Fig. E.4 Estructura MVC del proyecto de Intranet

Anexo F. Modificaciones de desarrollo

Este TFC se ha realizado en 4 fases de desarrollo diferenciadas por los plazos de entrega acordados entre J. C. Eurosit y su cliente. Como ya se explica en el apartado 1.2.4, en estos diferentes plazos de entrega se ha buscado cumplir con una serie de objetivos construyendo este desarrollo de una forma modular. Si bien es verdad que, al avanzar de una fase a otra, nos hemos encontrado en ocasiones con la necesidad de modificar ciertas partes de código que ya estaban desarrolladas en fases anteriores. En este apartado se explicará, fase a fase, las modificaciones que se han realizado a lo largo del desarrollo de este TFC.

- **Fase 1 – Implementación mediante lectura de etiqueta. Modo escáner. Planificado para la semana 10.**

  En esta fase inicial se define una BBDD muy sencilla funcional, incluyendo tan solo las tablas “defecto”, “ma_tipo_defecto”, “ma_linea” y “ma_modelo”. Respecto a las aplicaciones de escritorio, únicamente se desarrollan las primeras versiones de los programas Quality Checker y Quality Repair. En ambas, tan solo se establece el Modo Escáner, de tal forma que la lectura de los datos del asiento en curso se realiza mediante el escaneo del código QR de la etiqueta de secuenciación.

  En la figura 3.46 se puede visualizar un ejemplo de la aplicación Quality Checker desarrollada en esta fase.

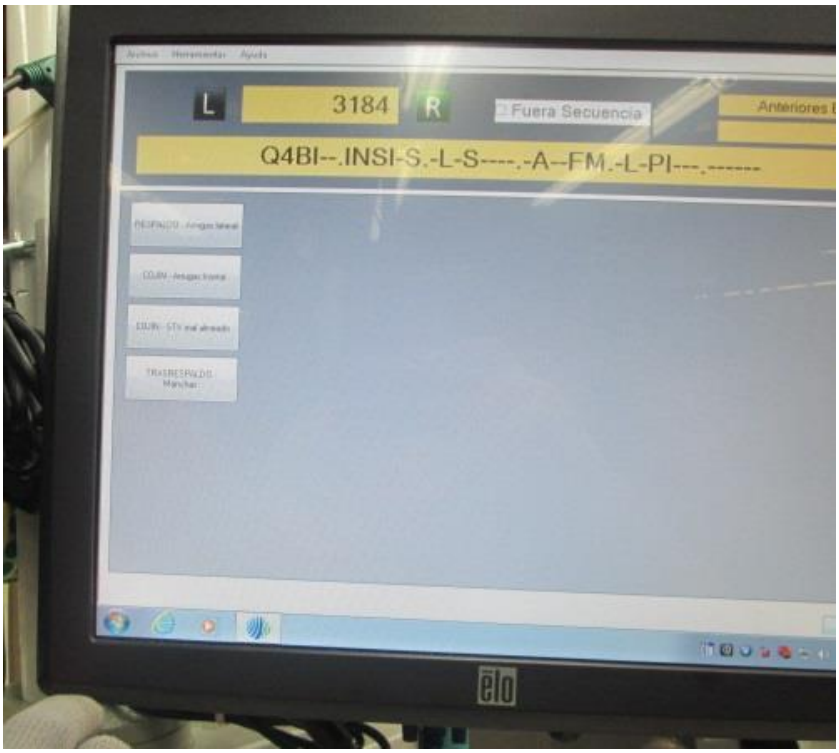


Fig. F.1 Quality Checker en su primera fase de desarrollo

- **Fase 2 – Implementación mediante lectura de PLC. Modo OPC. Planificado para la semana 16.**

  En esta fase de desarrollo se modifican las aplicaciones para conectar al PLC Siemens. Para ello se crean las etiquetas OPC de conexión a los datos de los PLCs de las líneas de montaje. A su vez se modifican las aplicaciones de escritorio para trabajar en Modo OPC. Se rediseña el interfaz gráfico de la aplicación estructurando los botones de defectos según categorías lo que supone incluir en la base de datos la tabla 'ma_familia' y redefinir la tabla 'ma_tipo_defecto' para agruparlos según familias. Se añade a su vez la funcionalidad de control de usuarios en las aplicaciones de escritorio, lo que implica la creación de la tabla "usuario" y la modificación de la tabla "defecto" para incluir el identificador del auditor de calidad que realice el registro de los defectos.

  En las figuras F.2 y F.3 se pueden visualizar un ejemplo de las aplicaciones Quality Checker y Quality Repair desarrolladas en esta fase.

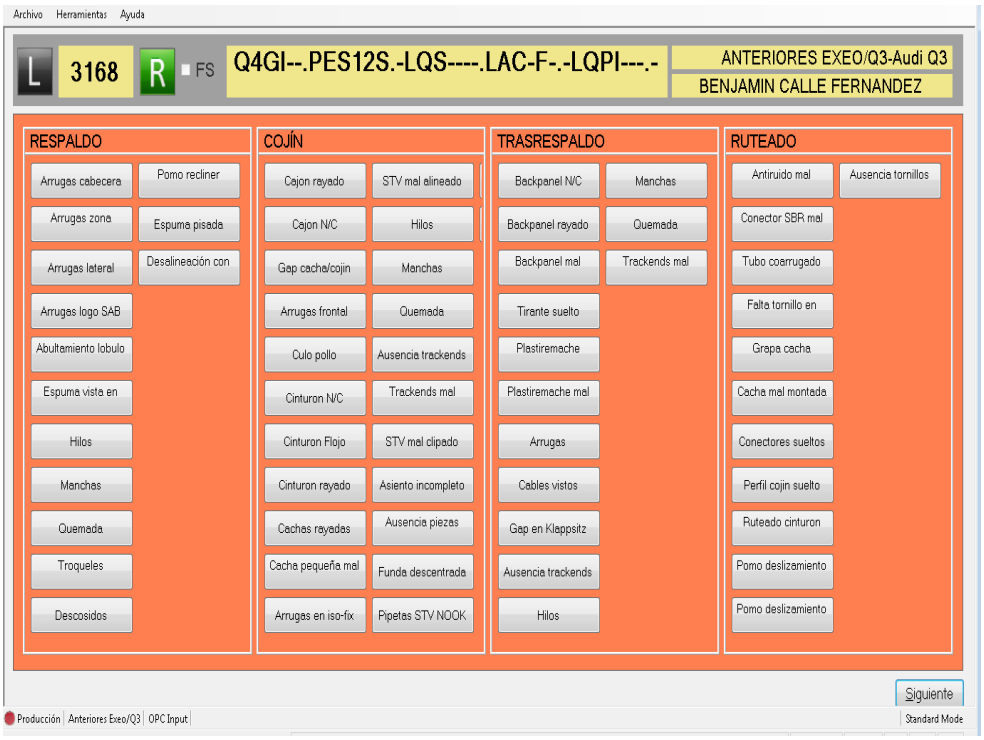


Fig. F.2 Quality Checker en su segunda fase de desarrollo

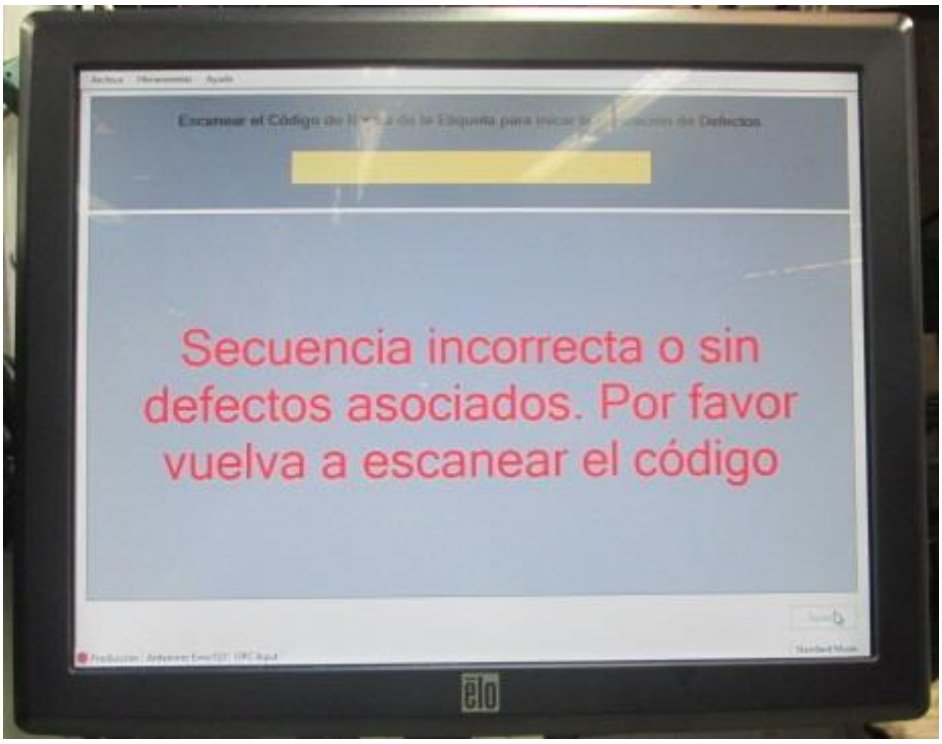


Fig. F.3 Quality Repair en su segunda fase de desarrollo

- **Fase 3 – Generación de informes y pantallas de información en tiempo real. Planificado para la semana 22.**

  En esta fase de desarrollo se crean nuevos procedimientos almacenados en la base de datos a los que accederán las aplicaciones web creadas. Se redefinen las tablas maestras de la base de datos incluyendo el campo habilitado para permitir dar de baja líneas de producción, modelos, tipos de defectos y familias sin necesidad de borrar sus registros. Se modifican a su vez los procedimientos almacenados para que únicamente los conceptos habilitados.

  Se diseñan y crean las tres páginas web de información en tiempo real y se instalan pantallas de 27 pulgadas a lo largo de las líneas de producción para mostrar los defectos registrados en tiempo real.

  Respecto a las aplicaciones de escritorio, se corrigen errores reportados por los auditores y operarios de producción en algunas funcionalidades que no suponen cambios relevantes.

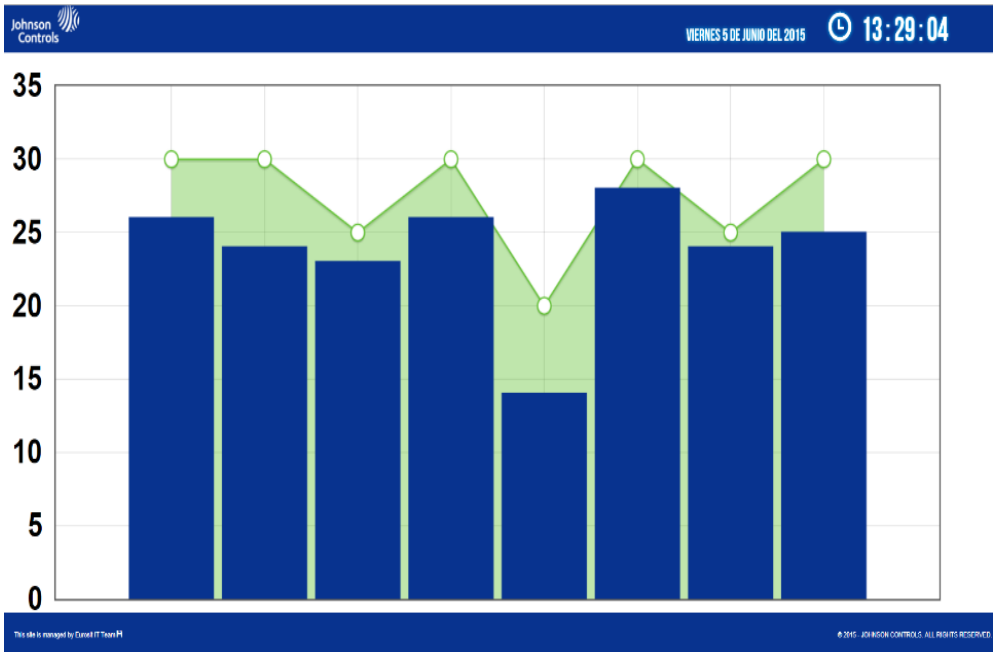


Fig. F.4 Producciones hora a hora por turno

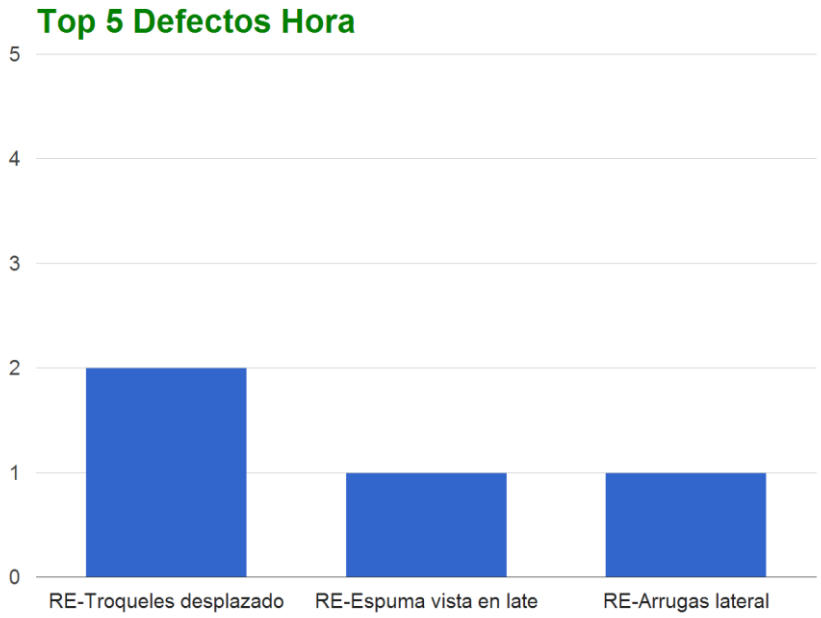


Fig. F.5 Top 5 de defectos acumulados en la hora en curso

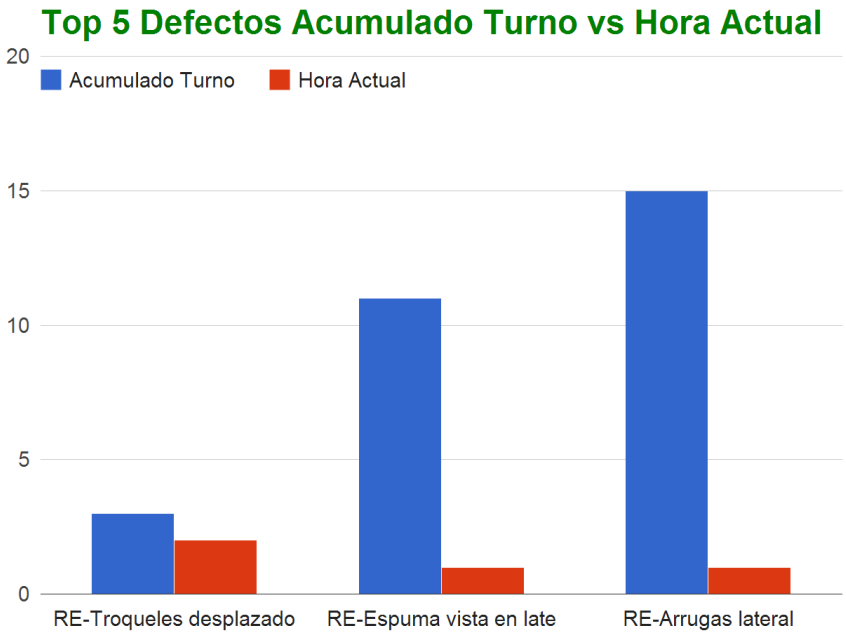


Fig. F.6 Top 5 de defectos acumulados por turno respecto a los defectos de la hora en curso

- **Fase 4 – Desarrollo de una aplicación para el nuevo puesto de inspección final. Planificado para la semana 25**.

  En esta fase de desarrollo se introduce un nuevo puesto de trabajo en el que se revisan otro tipo de listado de defectos. Esto implica la definición de nuevas etiquetas OPC correspondientes a los valores del PLC de este puesto. A su vez, se modifica la base de datos añadiendo la tabla 'ma_puesto_trabajo' y se redefinen las tablas 'ma_tipo_defecto', 'ma_familia' y 'defecto' para incluir esta nueva funcionalidad. También se crea la tabla 'usuario_puesto' en la que se relacionan los usuarios habilitados para trabajar en cada puesto de trabajo. Por último se modifican los procedimientos almacenados para adaptarlos a esta nueva funcionalidad.

  En cuanto a las aplicaciones de escritorio, se crea la tercera aplicación desarrollada en este TFC, Quality Firewall.

  Por último, se sustituyen las tres páginas web de información en tiempo real por una página que engloba todos los datos en una única vista.

  En las figuras F.7, F.8 y F.9 se pueden visualizar un ejemplo de las aplicaciones Quality Checker, Quality Repair y Quality Firewall desarrolladas en esta fase.

Fig. F.7 Quality Checker en su última fase de desarrollo

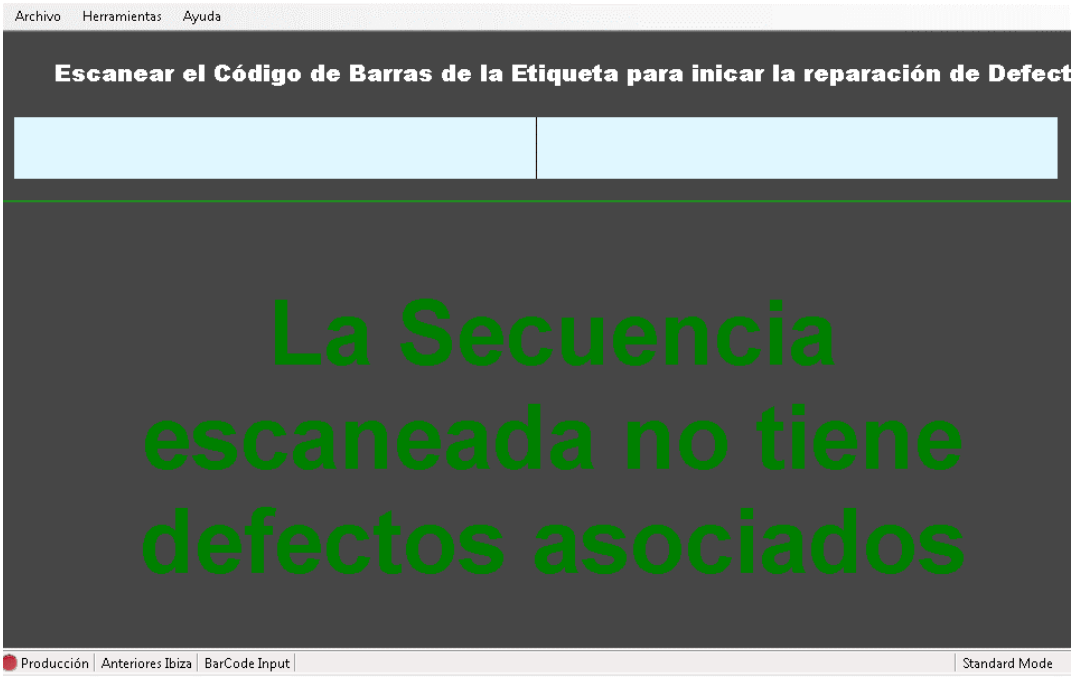


Fig. F.8 Quality Repair en su última fase de desarrollo

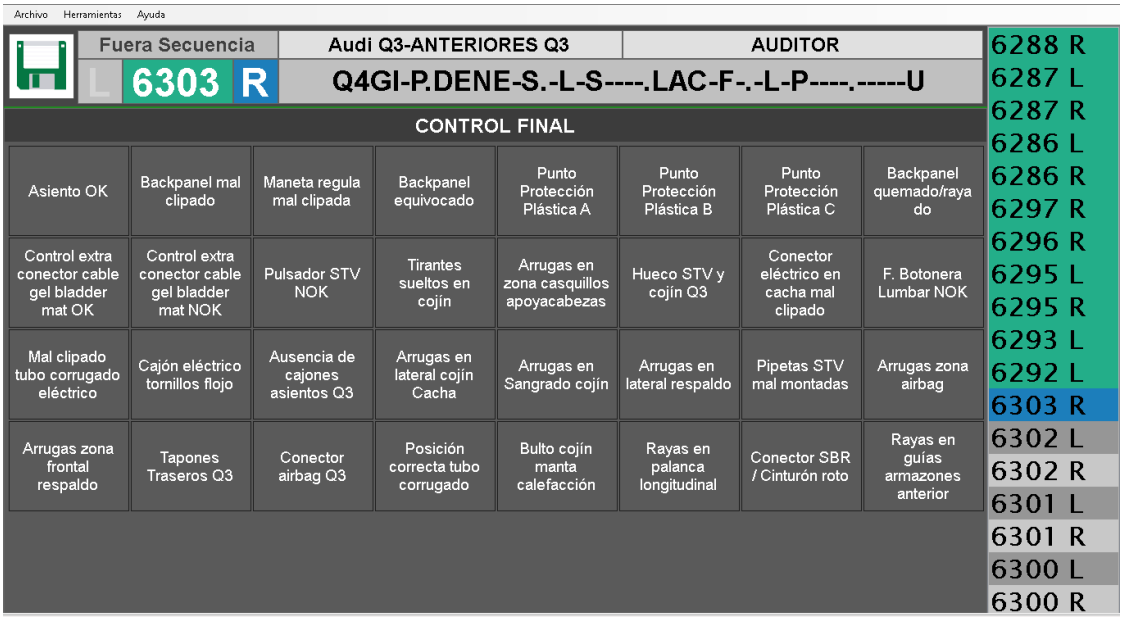


Fig. F.9 Quality Firewall en su última fase de desarrollo